Report No. FAA-RD-76-39

AD A 038543

Project Report

Discrete Address Beacon System

ATC-65

J. L. Gertz

The ATCRBS Mode of DABS

31 January 1977

Prepared for the Federal Aviation Administration by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.

1 Project Rept.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-76-39		
4. Title and Subtitle The ATCRBS Mode of DABS		5. Report Date 31 January 1977 6. Performing Organization Code
7. Author(s) J.L. Gertz		8. Performing Organization Report No. ATC-65
9. Performing Organization Name and Address Massachusetts Institute of Technology Lincoln Laboratory P.O. Box 73 Lexington, MA 02173	F19628-76-C-0002	10. Work Unit No. (TRAIS) Project No. 034-241-012 11. Contract or Grant No. DOT-FA72-WAI-261
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, DC 20591		13. Type of Report and Period Covered Project Report 14. Sponsoring Agency Code

15. Supplementary Notes

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology under Air Force Contract F19628-76-C-0002.

16. Abstract

The Discrete Address Beacon System (DABS) has been designed to be an evolutionary replacement of the present third generation Air Traffic Control Radar Beacon System (ATCRBS). Although the ATCRBS returns processed by DABS will be identical to those currently being employed, the DABS processing system will not merely mimic the present system. Instead, it has been designed to surpass current performance levels even while reducing the number of interrogations transmitted per scan. This will be made possible by utilizing the availability of several new features introduced by the DABS sensor. In particular, the employment of a monopulse antenna will permit both more accurate azimuth estimation with fewer replies per scan and improved decoding performance when garble is present.

The ATCRBS portion of the DABS sensor has been designed to be a complete, self-contained package that performs all ATCRBS functions required for aircraft surveillance. The major tasks it implements are:

1. Determining the range, azimuth, and code of each received ATCRBS reply
2. Grouping replies from the same aircraft into target reports and discarding fruit replies
3. Identifying all false alarm target reports due to reflections, coincident fruit, splitting, or ringaround
4. Initiating and maintaining a track on all aircraft in the covered airspace

The first function has been implemented in hardware while the remaining ones are performed in software. This report will discuss in detail only the software subsystems.

The ATCRBS system described in this report has been implemented in the ATCRBS Monopulse Processing System (AMPS) built at Lincoln Laboratory. Although the AMPS design is based upon the specifications contained in the DABS Engineering Requirements (ER), there are two major differences between AMPS and the ER system. First, the design described here is for a standalone ATCRBS system; no capabilities are built in to send, receive, or employ information from other sensors, and no formal interfaces to other ATC functions are defined. Second, this system was not intended to be a production prototype, so no reliability features have been included.

17. Key Words		18. Distribution Statement
Air Traffic Control ATCRBS DABS AMPS	Surveillance False Targets Tracking Radar	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	240	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

207650

Abstract

The Discrete Address Beacon System (DABS) has been designed to be an evolutionary replacement of the present third generation Air Traffic Control Radar Beacon System (ATCRBS). Although the ATCRBS returns processed by DABS will be identical to those currently being employed, the DABS processing system will not merely mimic the present system. Instead, it has been designed to surpass current performance levels even while reducing the number of interrogations transmitted per scan. This will be made possible by utilizing the availability of several new features introduced by the DABS sensor. In particular, the employment of a monopulse antenna will permit both more accurate azimuth estimation with fewer replies per scan and improved decoding performance when garble is present.

The ATCRBS portion of the DABS sensor has been designed to be a complete, self-contained package that performs all ATCRBS functions required for aircraft surveillance. The major tasks it implements are:

1. Determining the range, azimuth, and code of each received ATCRBS reply

2. Grouping replies from the same aircraft into target reports and discarding fruit replies

3. Identifying all false alarm target reports due to reflections, coincident fruit, splitting, or ringaround

4. Initiating and maintaining a track on all aircraft in the covered airspace

The first function has been implemented in hardware while the remaining ones are performed in software. This report will discuss in detail only the software subsystems.

The ATCRBS system described in this report has been implemented in the ATCRBS Monopulse Processing System (AMPS) built at Lincoln Laboratory. Although the AMPS design is based upon the specifications contained in the DABS Engineering Requirements (ER), there are two major differences between AMPS and the ER system. First, the design described here is for a standalone ATCRBS system; no capabilities are built in to send, receive, or employ information from other sensors, and no formal interfaces to other ATC functions are defined. Second, this system was not intended to be a production prototype, so no reliability features have been included.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF FIGURES (con't)

LIST OF FIGURES (con't)

THE ATCRBS MODE OF DABS

1.0 INTRODUCTION

The Discrete Address Beacon System (DABS) has been designed to be an evolutionary replacement of the present third generation Air Traffic Control Radar Beacon System (ATCRBS). The introduction of DABS sensors will proceed gradually over a number of years. The required changeover from ATCRBS to DABS transponders will occur long after the first DABS sensors are operational. Rather than incur the expense of requiring dual ATCRBS and DABS sensors at every DABS site, the DABS sensor has been designed to perform all necessary surveillance functions on both DABS and ATCRBS equipped aircraft.

Although the ATCRBS returns processed by DABS will be idencical to those currently being employed, the DABS processing system will not merely mimic the present system. Instead, it has been designed to surpass current performance levels even while reducing the number of interrogations transmitted per scan. This will be made possible by utilizing several new features introduced by the DABS sensor. In particular, the employment of a monopulse antenna will permit both more accurate azimuth estimation with fewer replies per scan and improved decoding performance when garble is present.

The ATCRBS portion of the DABS sensor has been designed to be a complete, self-contained package that performs all ATCRBS functions required for aircraft surveillance. The major tasks it implements are:

1. Determining the range, azimuth, and code of each received ATCRBS reply
2. Grouping replies from the same aircraft into target reports and discarding fruit replies
3. Identifying false alarm target reports which occur from reflections, coincident fruit, splitting, or ringaround
4. Initiating and maintaining a track on all aircraft in the covered airspace

The first function has been implemented in hardware while the remaining ones are performed in software. This report will discuss in detail only the software subsystems.

The output of the ATCRBS portion of the DABS sensor is target reports on ATCRBS equipped aircraft. Thus, the tracking function may appear to be unnecessary. However, the presence of internal track files is vital to the generation of accurate and complete target reports. Comparison of current scan reports with the previous scan information contained in the sensor track file permits the following types of report quality improvement to occur:

1. Incomplete aircraft codes can be completed

2. Suspected decoding errors can be identified

3. Reply correlation errors that produce incorrect mode associations can be identified and corrected

4. Coincident fruit, split, and ringaround reports can be suppressed

5. False target reports due to reflection can be identified and marked

The correlating track number for every target report is contained within the report.

An overview of all the functions performed by the ATCRBS system is presented in Figure 1-1. The remainder of this report will describe in detail the algorithms designed to perform these functions and the particular implementations of them developed by Lincoln Laboratory. For each algorithm, the rationale as well as the purpose will be presented in the hope that reader understanding will thereby be increased. The implementation presented here is felt to be efficient in terms of time and space and is intended to serve as a guide for other software designers. Clearly, alternate approaches exist.

The ATCRBS system described in this report has been implemented in the ATCRBS Monopulse Processing System (AMPS) built at Lincoln Laboratory. Although the AMPS design is based upon the specifications contained in the DABS Engineering Requirements (ER), there are several differences between AMPS and the ER system. First, the design described here is for a standalone ATCRBS system; no capabilities are built in to send, receive, or employ information from other sensors, and no formal interfaces to other ATC functions are defined. Second, this system was not intended to be a production prototype, so no reliability features have been included. Third, the confidence bit designations employed here are the exact opposites of the ER definitions. This is an historical problem that would be difficult to rectify internally, but which is trivial to overcome at the interfaces by simple bit inversion. Finally, many of the surveillance processing algorithms differ in minor respects from the ER rules. These reflect the increased knowledge that has been obtained through analysis of real-world data since the ER was written. These improvements will be included in future DABS ER revisions.

The AMPS system has fully implemented mode A and mode C processing capabilities, as algorithms for these modes are currently well defined. AMPS will also accept mode 2 replies if present and include them with each target report. Except that AMPS will attempt to associate the proper mode 2 code with each report, however, the presence of mode 2 is transparent to surveillance processing. In particular, no mode 2 code is maintained in the track file, mode 2 is not employed in any correlation decision, and no target report data editing decision is affected by the presence or absence of a mode 2 code.

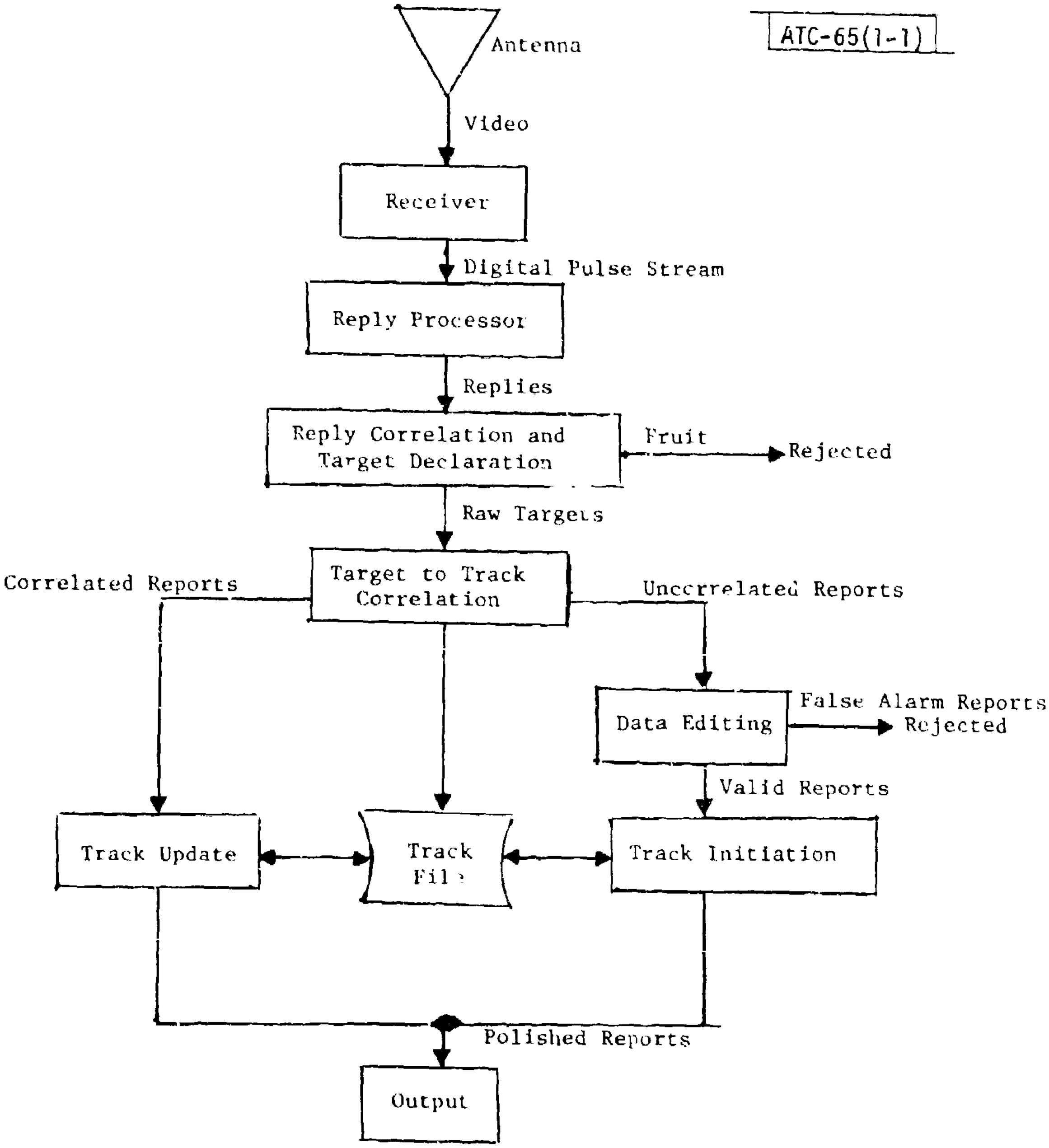


Figure 1-1: ATCRBS Portion of DABS Sensor

All the data structures in this report are drawn under the assumption the computer being used has 32 bit memory words. However, all fields have been designed to satisfy 16 bit boundaries, and thus a 16 bit computer could use the same structures directly (just by storing each 32 bit word in two 16 bit words). In fact, the ATCRBS implementation presented in this report has been programmed on both a 32-bit and a 16-bit computer.

The remainder of this report is structured as follows. Chapter 2 describes in overview the functions performed by the reply processor hardware and lists the inputs provided to the software. Chapter 3 presents a high level description of the software functions that are described in detail in the remainder of the paper, both to set them in perspective and to serve as a summary for readers not interested in the implementation aspects of the algorithms. Chapter 4 discusses the reply correlation and raw target formation procedures. The correlation of discrete code target reports and tracks is covered by Chapter 5. Chapters 6 and 7 present the more complex algorithms required for non-discrete correlation; the former chapter presents the preliminary target-to-track association function while the latter chapter presents the resolution of multiple association situations into the proper correlation pairs. The automatic initiation of tracks on new aircraft is described in Chapter 8, while the updating of these tracks from scan to scan is covered by Chapter 9. Chapter 10 then describes how various classes of false alarm target reports (reflections, coincident fruit, splits, or ringaround) are identified and processed. Finally, Chapter 11 discusses the use of primary radar reports in the ATCRBS system, both for reinforcing beacon reports and for providing surveillance for non-equipped aircraft.

2.0 REPLY PROCESSING

An ATCRBS reply, as illustrated in Figure 2-1, consists of between two and fifteen pulses. The function of the hardware reply processor is to identify all ATCRBS replies by searching the received pulse train for framing pulse pairs and then to decide which (if any) of the code pulses exist for each reply. The hardware also determines the range of each reply, from the time of arrival of the F1 pulse, and the azimuth of each reply, from the monopulse samples of all pulses received. The remainder of this chapter will highlight the key ideas of the reply processor design.

2.1 Reply Detection

A candidate ATCRBS reply is declared whenever two pulses separated by approximately 20.3 microseconds ("framing" pulses) are located in the input pulse stream. The candidate reply is accepted as a valid reply provided it meets both of the following criteria:

1. At least one of the framing pulses is declared to be received in the antenna mainbeam

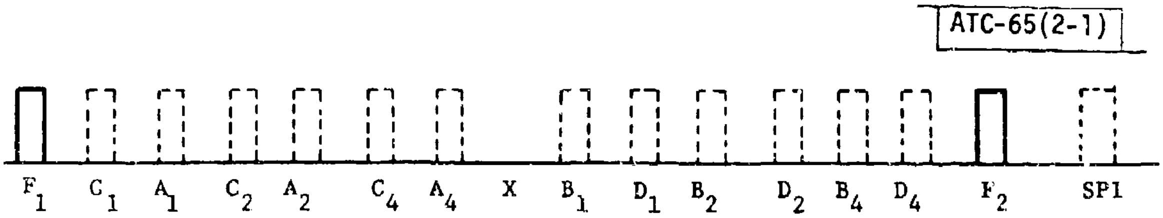
2. The reply is not thought to be a phantom

The first condition alludes to the fact that the ATCRBS processing hardware contains receive sidelobe suppression (RSLS) circuitry that identifies each pulse received in a sidelobe of the antenna. Thus sidelobe replies, which are not valid aircraft responses, can be eliminated.

A phantom reply is defined to be one created by pulses from two valid replies. As illustrated in Figure 2-2, when two replies overlap properly, a pulse of the first reply can be separated from one of the second by the 20.3 microsecond interval that characterizes framing pulses, thereby creating an intermediate candidate reply. The reply processor eliminates the middle reply whenever three candidate mainbeam replies are found whose relative times satisfy the phantom conditions.

Two other special types of replies, depicted in Figure 2-3, are identified by the reply processing hardware. The first, called a C_2-SPI phantom, occurs whenever a reply contains pulses in both the C_2 and SPI positions; since these positions are exactly 20.3 microseconds apart, they produce a phantom bracket. The other is the military identification reply, whose second half consists of a bracket whose F1 pulse is located in the SPI position of the real aircraft reply.

Clearly, two real replies from two different aircraft could produce either situation, so automatic elimination of either type of special reply is not permitted. Rather, azimuth correlation of the pulse in the SPI position



F_1, F_2 are framing pulses (always present), 20.3 μsec apart.

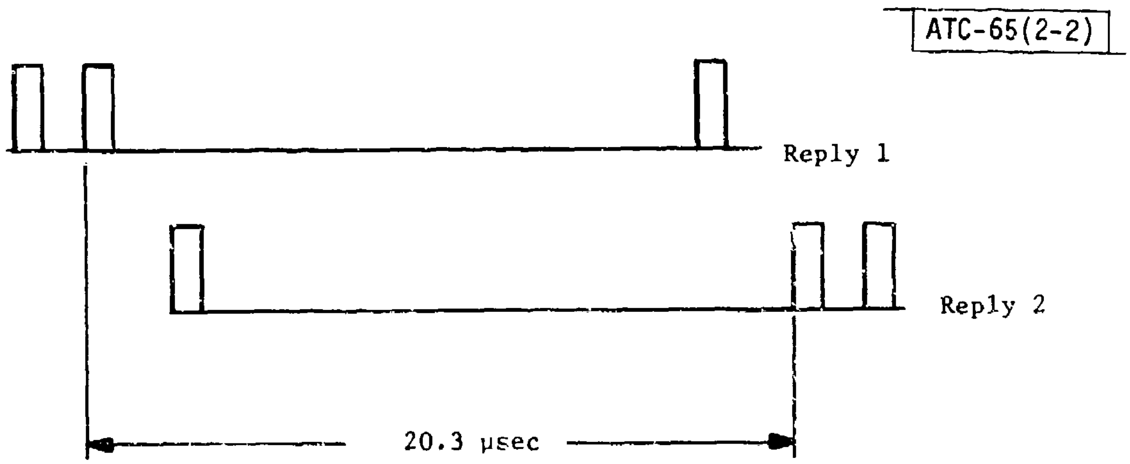
Each pulse is nominally 0.5 μsec wide.

Interpulse spacing is 1.45 μsec.

X position is normally unused.

SPI pulse, used for signalling, is 3 positions beyond F_2.

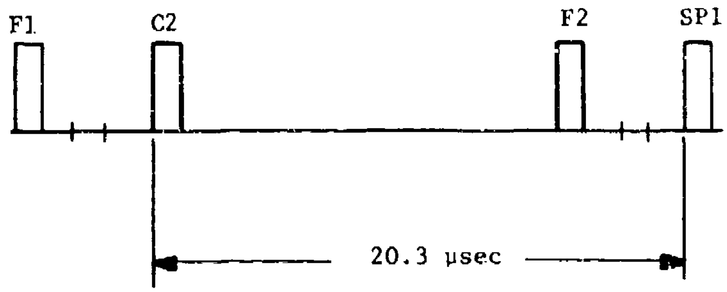
Figure 2-1: An ATCRBS Reply



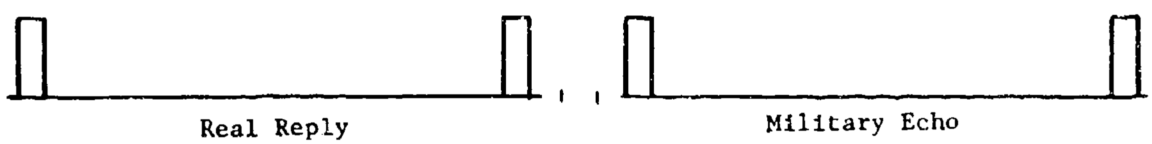
Pulse from Reply 1 and pulse from Reply 2 form a phantom bracket pair.

Figure 2-2: Creation of a Phantom Reply

ATC-65(2-3)



C2 and SPI pulses form a phantom bracket pair.



Special military reply mode, second bracket in SPI position of real reply.

Figure 2-3: Special Phantom Conditions

with the other pulse (C_2 for C_2-SPI phantom, F1 of first reply for military ID) is required. If correlation fails, the candidate reply is accepted, while if correlation succeeds, the reply will be discarded. The logic of these situations interacts with that of the normal phantom situation in such a manner that C_2-SPI phantoms must be eliminated immediately by the hardware, but military echos must only be marked by the hardware and later eliminated in software. That is, keeping C_2-SPI replies could result in the elimination of real replies as phantoms, while eliminating military echoes could result in phantoms being called real replies. Figure 2-4 clarifies this issue.

2.2 Reply Decoding and Confidence Bits

Once a reply has been detected, the reply processing hardware must determine, for each of the twelve code positions, whether or not a pulse exists in that position, and if so, whether or not it belongs to that reply (as opposed to another overlapping reply). This process is quite straightforward for a reply in the clear, but is difficult for a reply that is garbled by one or more other replies.

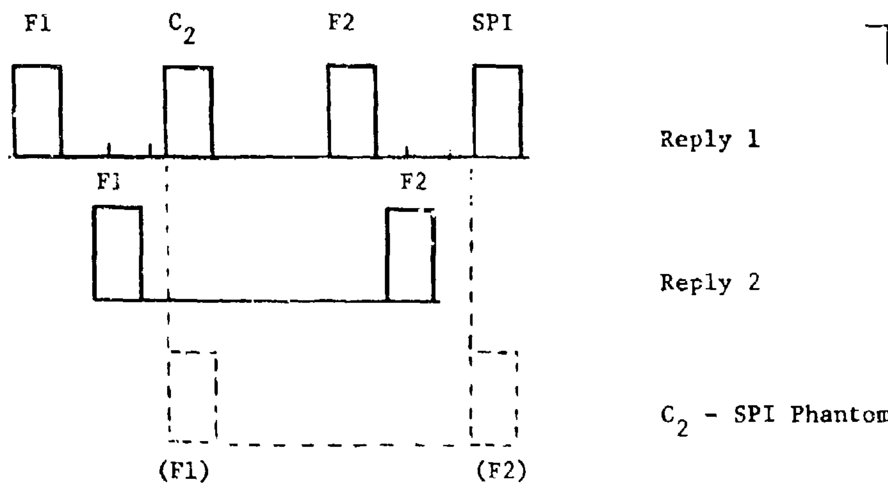
Since ambiguity is fairly common in garble situations, the reply processor may not be able to decide whether or not a specific code pulse for a given reply is present. Rather than force a possibly wrong guess to be made, the idea of confidence flags was developed. For each code bit decision, a corresponding confidence decision, high or low, is made. When the decision is straightforward, the confidence flag is turned off ('0'); when the decision is ambiguous, the best guess is made, but the confidence flag is set ('1'). The important point that will be seen later is that only high confidence code bits will be employed in any of the code comparison tests.

The rules for determining what values of code and confidence to assign to a given pulse position of a given reply are the following:

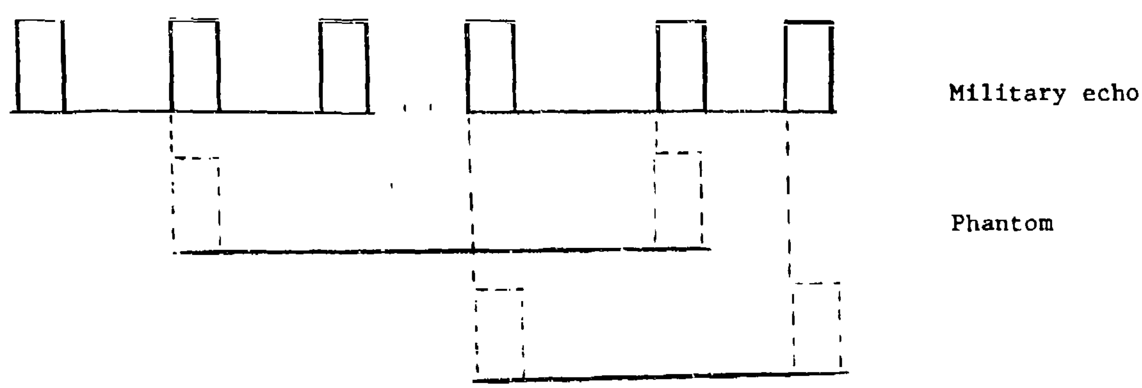
H0: a high confidence 0 is declared whenever no pulse is detected in the code position

H1: a high confidence 1 is declared whenever a mainbeam pulse is detected in the code position that correlates in azimuth with the reply reference azimuth and fails to correlate with the reference of every other garbling reply (if any)

L0: a low confidence 0 is declared whenever either (a) a sidelobe pulse is detected in the code position, or (b) a mainbeam pulse is detected that fails to correlate in azimuth with the reply reference but succeeds in correlating with the reply reference of a garbling reply



If C_2 - SPI phantom is kept, reply 2 is eliminated as the "reply in the middle".



Phantom can be eliminated as "reply in the middle" only if military echo reply is maintained.

<u>Figure 2-4:</u> Phantom Logic Interactions

L1: a low confidence 1 is declared whenever a mainbeam pulse exists in the code position that either (a) fails to correlate in azimuth with the reply reference and with the references of all other garbling replies (if any), or (b) correlates successfully with both the reply reference and the reference of one or more garbling replies

An example of the application of these rules in a garbling situation is presented in Figure 2-5.

The reference azimuth for a reply is initially set to the azimuth of the F1 framing pulse of the reply. However, if this pulse is located in a garble region, the azimuth of the F2 pulse is utilized.* The reply reference azimuth is updated each time a high confidence 1 is declared for the reply (code pulse or framing pulse) through simple averaging of the old reference with the new sample. If the initial reference azimuth is not confirmed by a succeeding pulse, the azimuth of the reply is defaulted to the antenna boresight and a special marking is set.

2.3 Reply Processor Outputs

For each interrogation sweep, the reply processor transmits to the ATCRBS software the following two items of information:

1. Mode of the sweep (A, C, or 2)
2. Antenna boresight azimuth

In addition, for each reply declared by the reply processor, the following set of information is provided:

1. Reply range
2. Reply boresight azimuth
3. Final reply monopulse reference
4. Reply code
5. Reply code confidence
6. Special implementation dependent reply attributes

---

*It should be noted that these reference azimuth selection rules permit a sidelobe pulse to be chosen. A modification being made to the DABS reply processor implementation corrects this oversight by discarding any reply each of whose framing pulses is either garbled or sidelobe. In the AMPS implementation, this rule change is being handled in the reply correlation software, as described in Chapter 4.

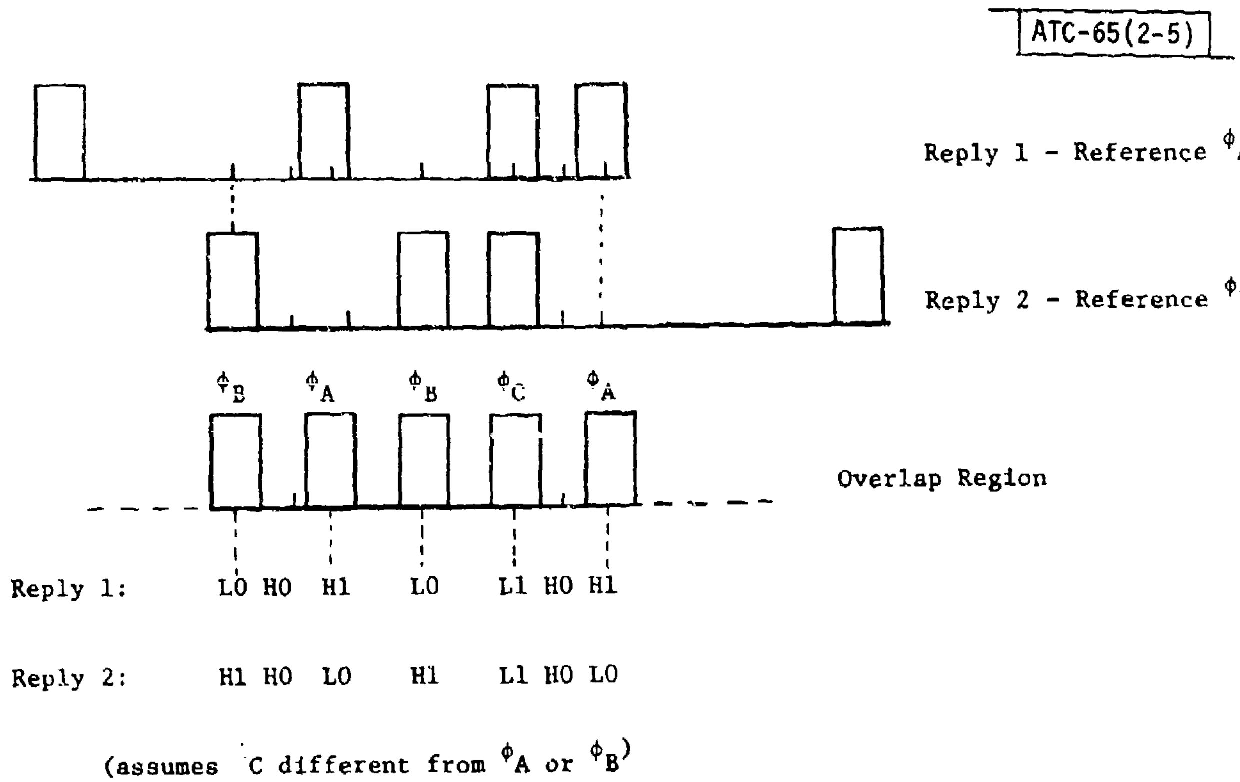


Figure 2-5: Confidence Bit Decisions

The range is given in time counts from sweep interrogation until reception of the F1 pulse, and hence it must be converted to miles at some later time. The reply boresight azimuth is the antenna azimuth at the time the reply was received. After the reply correlation software determines the off-boresight azimuth of the reply, by a table lookup whose index is the final monopulse reference value, the two azimuth values are summed to produce the actual reply azimuth. The code and code confidence bits are ordered as follows:

$$A_4A_2A_1B_4B_2B_1C_4C_2C_1D_4D_2D_1F_1F_2XSPI$$

where the F1 and F2 bits are optional. The format of the reply block transmitted by the AMPS reply processor, and the list and definitions of all the special reply attributes it provides, are provided in Figure 2-6.

0–15	16–31
Range	Code

0–13	14–15	16–31
Boresight Azimuth	00	Confidence

0–3	4–10	11	12–15	16–19	20–23	24–31
Mode	Not Used	(1)	Decoder	Not Used	(2)	Final Monopulse Reference

0–5	6–7	8–15	16–19	20–31
Not Used	(3)	Special Monopulse Check	Number of Pulses	Total Monopulse Accumulation

Notes:

Range: least significant bit = 60.4 nsec

Code: $A_4A_2A_1B_4B_2B_1C_4C_2C_1D_4D_2D_1$ Fi F2 X SPI

Azimuth: least significant bit = .022°

Confidence: same order as code, "0" = high confidence

Monopulse Fields: counts

(1) Test Bit

(2) $N \geq 2$ bit - number of confirming monopulse samples
Reference - F1 or F2
Sidelobe status of reference
SPI presence

(3) Overload bit
Sweep header bit

ATC-65(2-6)

Figure 2-6: Reply Processor Input Block

3.0 THE ATCRBS SOFTWARE SUBSYSTEM

The ATCRBS software subsystem consists of two separable components: reply correlation and surveillance processing. The first program, which is executed once per sweep, attempts to group replies from the same aircraft into raw target reports and to reject fruit replies. These target reports are then processed once per sector (nominally 11.25°) by the second program, which corrects, completes, and labels the reports through reference to track history information. Since these two programs interact solely through a one-way transfer of target reports, they can easily be implemented in separate computers if so desired. This chapter will discuss the algorithms for both components in summary fashion, while later chapters of this report will give the implementation details. Thus, a reader may refer to the corresponding chapter for any topic on which he desires more information. Figures 3-1a and 3-1b present a flowchart of the overall ATCRBS software subsystem that is described herein.

Although the basic functions to be performed by this system are identical to those of the current ARTS and NAS systems, it will become apparent that the algorithms required to implement them often differ considerably in method and complexity from existing ones. The main reason for these changes is that significant differences exist between the target reports of the current ATCRBS system and the one proposed as part of DABS. This fact becomes clear when one considers the following table:

Attribute	ARTS	DABS
typical runlength	16	4
garble bits	1	12
azimuth	beamsplit	monopulse

The long runlength in ARTS helps to prevent extraneous reports (fruit correlation, code splits, azimuth splits) from being declared. DABS raw reports, on the other hand, are often extraneous or contain code errors due to the very short runlength. Thus, data editing, and the compilation of the track files to support it, are necessary features of surveillance processing for DABS data.

Since ARTS reports contain only one garble bit (indicating clear or garbled code) and have an azimuth declared through beamsplitting, it is not surprising that the report quality is often low in crossing situations. Thus, to prevent track swaps, correlation is often not attempted in ambiguous situations. DABS reports, on the other hand, contain a garble bit for every code bit. Even in severe synchronous garble, some part of the report code will be known with certainty. This fact, combined with the accuracy of a monopulse azimuth, justifies attempting correlation in all situations. As a result, the correlation algorithms presented in this paper are far more complex than those currently employed. The resulting system performance, based on tests with live data, strongly indicates the added features are worth their cost.

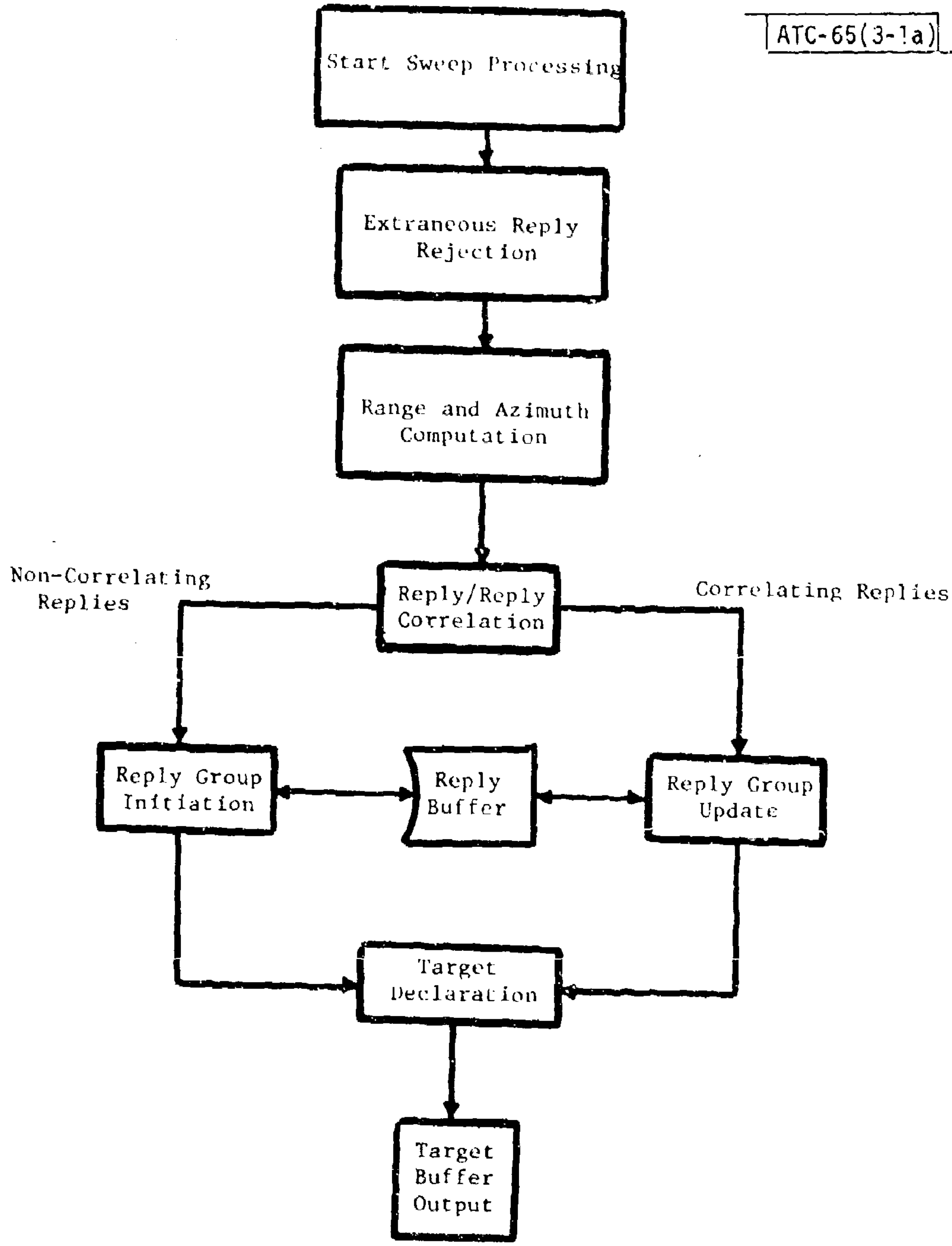


Figure 3-1(a): Reply Correlation Flowchart

ATC-65(3-1b)

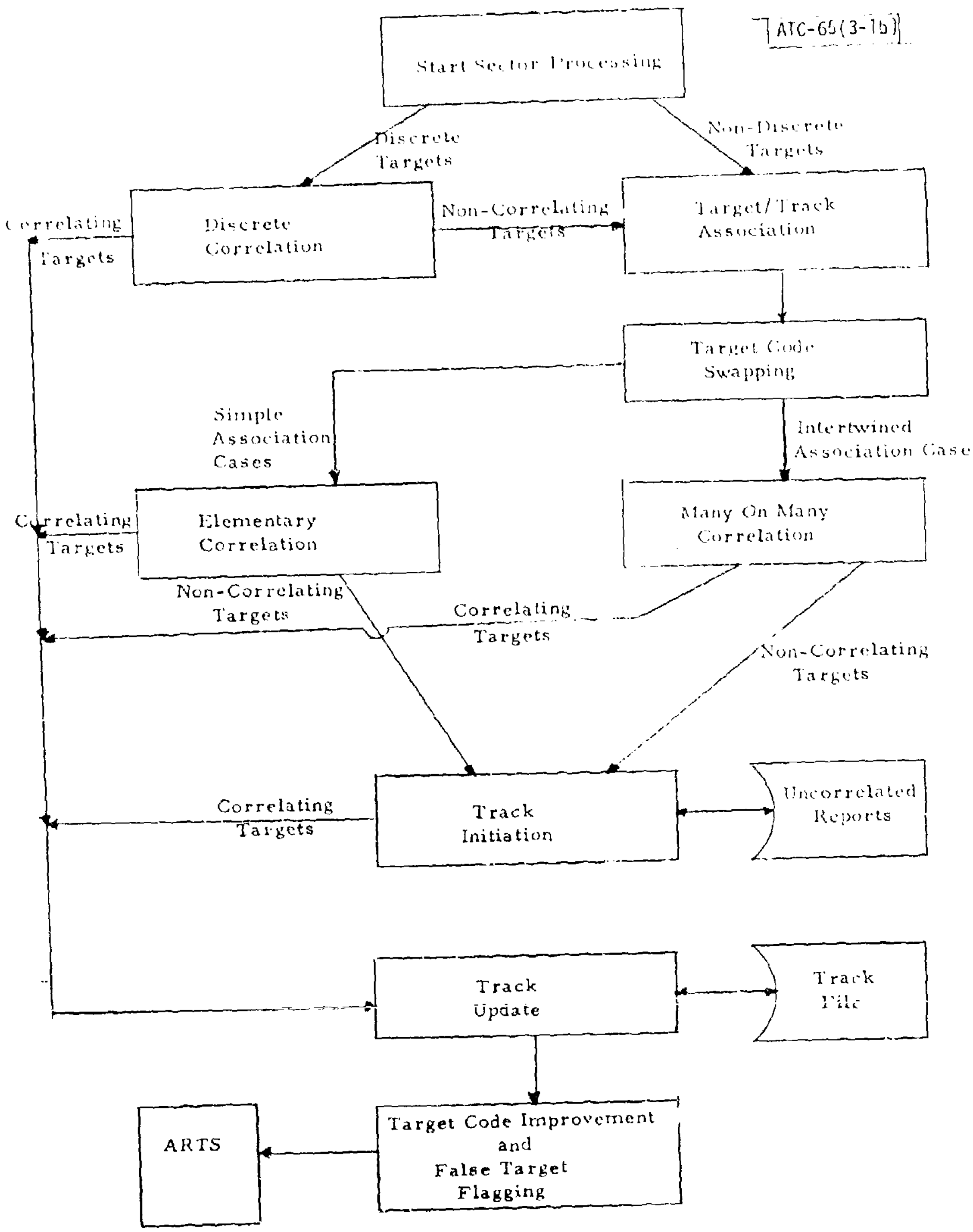


Figure 3-1(b): Surveillance Processing Flowchart.

3.1 Output Reports

The primary output of any ATCRBS sensor is a stream of target reports, hopefully one per scan for each aircraft in the coverage region. In a DABS sensor (and in the AMPS implementation of it), two types of reports exist: raw and polished. Raw reports are those declared through reply correlation. They are often incomplete in their information fields, and on occasion are due to false alarms rather than to real aircraft. Polished reports, on the other hand, have been processed through several software improvement algorithms that make use of track history information. Those reports felt to be valid are completed and labelled with a track file number, while those thought to be false alarms are discarded. In normal circumstances, only reports of the former type are output to the ATC users.

DABS reports are output to other DABS sensors, to the Intermittent Positive Control (IPC) function, and to various ATC users. AMPS reports, however, are only output to one ATC function at any time. The format of these reports is dependent upon which user (ARTS, NAS, etc.) is active. To indicate the ensemble of information available to any user, Figure 3-2 presents the final internal format for a report ready to be output. The special purpose bits, as indicated, are used for output screening, special report flagging, and analysis aids.

In the normal case, a target report is output in the same azimuth sector in which it is received. However, when target to track correlation requires future information to make its decision, the report may be delayed in the system. The maximum number of sectors that a report may be so held before being output is a system parameter. When the limit is reached, correlation is performed whether or not additional information is possible.

3.2 Reply Correlation and Target Formation

At the end of each sweep, after all replies have been received from the reply processing hardware, each is checked to see whether it was caused by a characteristic ATCRBS system problem rather than by a legitimate aircraft response. Examples of such effects that generate extraneous replies are sidelobe/mainbeam garble, military echoes, and out-of-specification (wide pulses) transponders. All such replies are eliminated. Remaining replies have their range and azimuth estimates computed by the software from the time and monopulse information provided by the hardware.

The reply correlation function then processes each acceptable reply in an attempt to correlate it with replies received on previous sweeps. This search is aided by a reply sort table, which permits identification by range of all existing reply groups (either uncorrelated replies or unions of two or more correlated replies). The new reply is correlated with the first group found for which the following four conditions are satisfied:

0 15 16 31

<table>
<tr><td colspan="2">Range</td><td colspan="4">Azimuth</td></tr>
<tr><td colspan="2">Mode A Code</td><td colspan="4">Mode A Confidence</td></tr>
<tr><td>Altitude</td><td>Unused (4)</td><td colspan="3">Altitude Confidence</td><td>Altitude Type (4)</td></tr>
<tr><td colspan="2">Special Bits</td><td># A Replies (3)</td><td># C Replies (3)</td><td colspan="2">Correlating Track No.</td></tr>
<tr><td>Mode 2 Code</td><td># 2 Replies (4)</td><td colspan="3">Mode 2 Confidence</td><td>Time in System (4)</td></tr>
</table>

(4) means 4 bits in the field

<u>Special Bits:</u>	<u>Reference Section</u>
Test Target	-
Edited Out Target	10.3 - 6
False Target	10.1 - 2
Boresight Target	8.2
Radar Reinforced	11.1
Code In Transition	9.1
Potential Swap Target	4.6
Swap Performed	6.4
Reconstructed Target	5.2
Discretely Correlated	5.3
Velocity Reasonableness Used	6.3
2-On-2 Or Many-On-1 Case	7.4
Many-On-Many Case	7.5
Deviation Score Used	7.2
Turn Detected	9.3
Code Improved	9.1

ATC-65(3-2)

<u>Figure 3-2:</u> Final Target Report Format

1. The range difference between the reply and the group is no greater than $\Delta\rho_{max}$

2. The difference between the monopulse azimuth estimates is no greater than $\Delta\theta_{max}$ (if either the reply or the group contains only an uncorrected boresight estimate, due to a default condition, this test is waived)

3. The group has not already correlated with another reply from the current sweep

4. The code of the reply agrees with that of the group (waived for mode 2)

If a successful match is obtained, the new reply attributes are combined with those of the existing reply group to produce an updated group specification. Otherwise, the reply is sorted into the range sort table and becomes available for correlation with future sweep replies.

After all replies from the current sweep have been processed, reply groups that are known to be complete, based on the number of sweeps that have occurred since the oldest reply contained within them, are converted into raw target reports. As part of this conversion, the mode C code is translated into altitude flight level. These reports are collected in a buffer, and once per sector are passed as input to the surveillance processing algorithms.

Ordinarily, only groups that contain two or more replies of modes A and C are made into raw target reports. However, any uncorrelated mode A or C reply located spatially near any other reply group will be turned into a special 1-hit raw report. Such reports, as explained below, are intended for use in code swapping to correct reply correlation errors.

3.3 Discrete Code Correlation

The ATCRBS system employs two types of identity codes, discrete and non-discrete. Discrete codes are assigned uniquely to aircraft within a single control area, while non-discrete codes can be used by all aircraft in the same flight class (such as descending IFR). Thus, agreement in mode A code between a discrete target report and a track is generally sufficient for target to track correlation, while more complex criteria are required to correlate non-discrete targets and tracks.

All ATCRBS track data, for both discrete and non-discrete tracks, are physically located in the same track file. However, a separate hash coded table permits all discrete code tracks to be accessed through their code. Thus, whenever a discrete code target report is to be correlated, it is possible to determine whether or not a track possessing the same code currently exists.

A target report and a track having the same discrete code are correlated whenever both of the following conditions are met:

1. Only one track exists with that code (assignment failures or tracking errors could produce duplication)

2. The target and track associate in range, azimuth, and altitude

Only target reports that possess no low confidence code bits are considered discrete; reports with discrete codes that have some uncertainty must be treated as non-discrete reports.

3.4 Target to Track Association

The first step in correlating non-discrete target reports, or discrete reports not successfully correlating as above, is to determine all possible pairs of target and track associations. From these pairs, the best correlations will be selected in the manner described in the next subsection. As part of the association process, many reply correlation and reply processor errors will be corrected through a process called code swapping.

As a minimum condition for association, a target report and a track must lie close together in range and azimuth. Three association zones are defined around each track for this test. These zones, denoted by 1, 2 and 3, correspond to expected prediction errors for aircraft flying straight, turning, and maneuvering in an unusual manner respectively.

In addition, code and altitude compatibility are checked for each potential association pair. If both modes agree, the association is accepted, while if neither mode agrees, the association is rejected. Zone 1 or 2 situations in which only one mode agreement exists are processed by the code swapping algorithm, which identifies and corrects cases of improper mode pairing by the reply processor.

Two target reports swap their mode A codes whenever a situation satisfying all of the following criteria is identified:

1. The reports are within the reply correlation range and azimuth windows of each other

2. No nearby track possesses the mode A and C pairings resident in either report

3. There exists a track that possesses the mode A code of one report and the mode C code of the other report

The reply correlation error that produced these improper mode pairings could have been caused by two aircraft crossing, by a high confidence bit error in the reply processor, or by the existence of a nearby fruit reply. In the first instance, code swapping will produce two proper reports, while in either of the latter two cases code swapping will replace the erroneous code with the correct code. The correct code has been maintained, since even if the reply containing it were uncorrelated, the reply correlation rules would have created a 1-hit report. Figure 3-3 illustrates the formation and resolution of two typical code swap situations.

If a report/track association pair with agreement in only one mode resulted in code swapping, the new pair, with both modes in agreement, is accepted. If no code swapping was possible, the pair is accepted if altitude agreement exists and rejected otherwise. This rule reflects the fact that identity codes can change from scan to scan, while large altitude changes are impossible.

Finally, if any accepted association pair is suspect, either by being in zone 3 or in zone 2 with a mode disagreement, a velocity reasonableness test is made. This test rejects all associations in which it is physically impossible for the aircraft under track to be located at the target report position.

3.5 Target to Track Correlation

Once all the target/track association pairs have been identified for a sector, a determination of the "correct" target report for each existing track must be made. Two types of scoring mechanisms are employed in this procedure to rank the various pairings: the Quality Score and the Deviation Score.

The Quality Score for a target-to-track association measures the degree to which the characteristics of the target report match those of the track, as well as the degree of certainty as to the validity of the report and track (that is, that they correspond to real aircraft and not system errors). The decision items that constitute this score, in order of decreasing importance, are the following:

1. Association zone (1, 2, or 3)

2. Mode A code agreement

3. Number of replies in the report

4. Mode A confidence of the report

5. Mode C altitude agreement

6. Track validity

Example 1:

2 aircraft at nearly the same range

Sweep	Mode	Order of reply reception	
1	A	A_1 A_2	
2	C	C_2 C_1	(due to different inter-mode delays)
3	A	A_1 A_2	
4	C	C_2 C_1	

Report	Before Swapping	After Swapping	
1	A_1, C_2	A_1, C_1	both valid
2	A_2, C_1	A_2, C_2	

Example 2:

fruit reply near aircraft

Sweep	Mode	Order of reply reception
1	A	A
2	C	C_f C
3	A	A
4	C	C

Report	Before Swapping	After Swapping	
1	A, C_f	A, C	
2	-, C	-, C_f	2nd discarded

Figure 3-3: Code Swap Utilization

The Quality Score is computed by evaluating where the target and track attributes fall on the scale of values defined for each item, and then weighting and summing these individual scores. The lower the resulting score, the better the association.

The Deviation Score for an association measures the detailed geometrical relationship between the target and track positions. Both the magnitude and direction of their difference is considered. Due to the complexity of these calculations, the Deviation Score is employed only when the Quality Score utilization results in a tie between two association pairs.

The correlation procedure for a track has two interrelated components: determining the best target report for the track, and deciding whether or not to postpone the correlation decision. The decision could be delayed whenever the track's association box extends beyond the current sector, giving it a reasonable expectation of finding a superior target report in a subsequent sector. When the decision to postpone is made, the track and all of its associating reports are carried over into the next sector for reprocessing.

If the association box of the track includes sectors prior to the one in which its prediction lies, association will begin prior to that sector. The track will not be permitted to correlate, though, before targets from its predicted sector have been received, as that is where the correct target is most likely to occur. Once the targets from the predicted sector have been received, correlation for the track will be attempted. If a correlating target is identified, the correlation will be accepted provided at least one of the following three conditions is met:

1. The Quality Score is lower (i.e. better) than a specified value

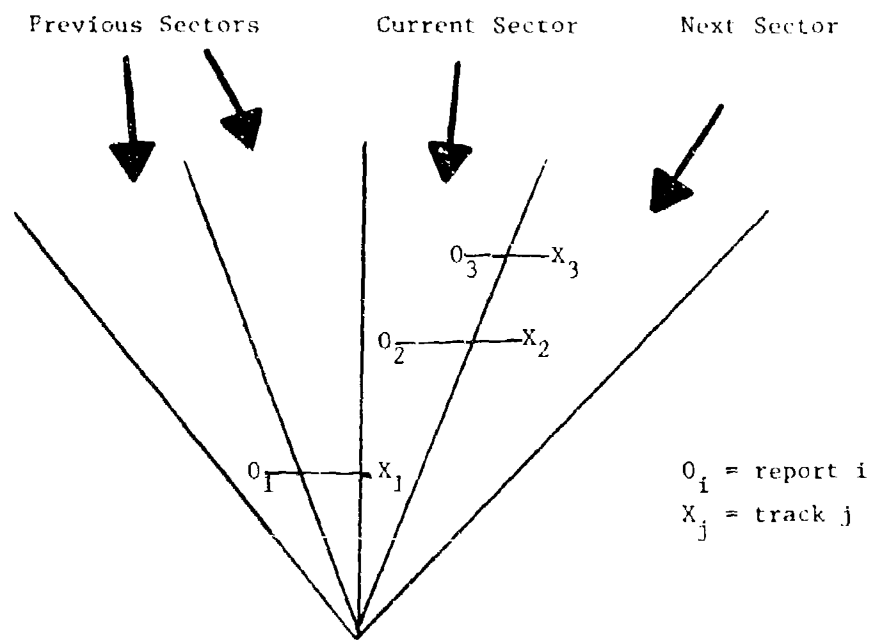
2. The target is not permitted to be delayed any longer in the system

3. The track has already received all possible associating targets

If none of these conditions is satisfied, correlation for the track is postponed for another sector. Figure 3-4 demonstrates the application of these rules in a typical situation.

The algorithm for determining the best associating target report for a track depends upon the complexity of the associative system linkages. If one track and one report associate only with each other, that report is selected. If several reports associate only with one track, the report with the lowest Quality Score is selected. In case of a tie, the Deviation Scores are employed as tiebreakers. An analogous dual rule is used when several tracks associate only with one report. Finally, when a many-track-many-report associative system exists, the pairings that minimize the sum of the selected Quality Scores are chosen. The algorithm that performs these selections is a best first approximation to the optimum solution of the assignment problem.

ATC-65(3-4)



Assume a report can be delayed at most 2 sectors

Correlation 1: accepted, as report cannot be held any longer

Correlation 2: accepted if score is good enough, else delayed

Correlation 3: delayed automatically, as track is predicted into a subsequent sector

Figure 3-4: Correlation Timing Rules.

3.6 Track Initiation

A new ATCRBS track is initiated whenever uncorrelated target reports are found on two successive scans that appear to have been generated by the same aircraft. The criteria that are employed in making this judgment are that the two reports:

1. Be sufficiently near each other that a real aircraft could traverse the distance in one scan

2. Agree in identity code

3. Be close together in altitude

Whenever two reports are found that satisfy these conditions, a new track file entry is created and placed on the list for the current sector. In addition, if the identity code of the track is discrete, the track is entered into the discrete track hash code table to permit future discrete code correlations.

Two distance zone sizes are used for the first test, corresponding to normal and exceptional aircraft respectively. If the search based on an uncorrelated target on the present scan locates one or more satisfactory reports from the previous scan that fall within the first zone, new tracks are initiated for all such cases but no tracks are begun for pairs that require the larger zone. If no first zone situations are found, however, tracks are started for reports located in the second zone.

Although a single uncorrelated target report can initiate more than one new track by the above procedure, it is clear that only one of these tracks can correspond to a real aircraft. The valid track in this group should be the only one to correlate on the subsequent scan. To permit the immediate dropping of the other phantom tracks, all tracks initiated by the same report are linked together. Then, when one of the set correlates and the others fail, these latter tracks can be identified and eliminated from the system.

3.7 Track Update

After the target to track correlation process has been completed for a sector, all tracks which have had their correlation resolved, either successfully or unsuccessfully, are predicted forward to the next scan. Those tracks whose correlation decision was postponed, and hence have not completed the correlation process, are not updated at this time. All tracks initiated during the current sector are automatically predicted ahead.

Tracks that possess correlating target reports, including newly initiated tracks (whose correlating report is the one that led to its formation), go through a two-step range and azimuth updating procedure. First, the current

predicted position and velocity are adjusted to reflect the location of the correlating target report. For a general α, β tracker, this smoothing would be a compromise between the prediction and the data point positions. At present, however, the ATCRBS system employs a 2-point tracker. This means that the smoothed position becomes that of the correlating report and the smoothed velocity is determined totally by the last two such reports. After the track is smoothed, the new velocity estimate is used to predict the track position ahead one scan.

In general, ATCRBS tracking is done in ρ, θ coordinates. However, if the track comes near the sensor, improved prediction equations are required in order to minimize curvature errors. For moderately close tracks, second order ρ, θ prediction is employed; for very close tracks, exact X-Y prediction is used. The rationale for not using X-Y prediction at all ranges is that the coordinate conversion required for target reports is very time consuming, while the system gain at other than close ranges is negligible.

The identity code and altitude fields of a correlated track file are also updated each scan. In general, the target code will agree with that of the track, so no code modification action is required. However, if the track is initiated in garble, several scans may be required to construct the entire code. Also, the code of an aircraft could change from time to time due to controller direction. The altitude update simply keeps the most current altitude of the aircraft in the track file.

After a track has been updated, the sector in which it should first appear on the next scan must be computed. This is done by centering a standard zone 3 sized correlation box at the new predicted position. The sector that contains the smallest azimuth value included in this box is the one sought. The track is then placed on the linked list of tracks for that sector and will be available to begin its next correlation process when that sector is next encountered.

Tracks that fail to correlate must also be updated, although the procedure is somewhat different. First, if the track has failed to correlate for a specified number of consecutive scans, it is dropped. An exception to this rule is made whenever the track is passing through the cone of silence of the sensor. In addition, since no report is present, no smoothing of the track position, nor identity code or altitude update of the track, can be made. The mechanism used to predict ahead a coasted track is identical to that for a correlated track, as is the method for determining the sector in which to place the track. However, the size of the correlation box employed in this latter calculation is larger, as its size grows with each coast to reflect the increasing uncertainty in the actual aircraft position.

3.8 False Alarm Target Reports

Not every raw target report created by the reply correlation process corresponds to a real aircraft position. Several inherent properties of the ATCRBS system will produce various types of false alarm target reports. To the extent possible, the surveillance processing subsystem attempts to identify and eliminate these reports.

The four types of false alarm reports specifically handled by the software are:

1. False targets - produced by replies bouncing off reflecting surfaces

2. Fruit targets - produced when fruit replies coincidentally correlate

3. Split targets - produced by the failure of reply correlation to group together all replies emanating from an aircraft

4. Ringaround targets - produced by sidelobe replies which were not suppressed

When any of these reports are identified, the system will take the action specified by the user. The alternatives he can choose are : (1) immediate elimination of the reports, (2) marking the reports and not allowing them to be used in correlation or track initiation, or (3) marking the reports but otherwise processing them in the normal manner. If the third alternative is selected, any tracks initiated by false alarm reports will also be marked as false.

False targets are generally caused by the reflection of aircraft responses off buildings, hangars, or other structures near the sensor, thereby causing an apparent aircraft position behind the reflector. Depending upon the size of the reflector, such false targets may persist for several scans and initiate false tracks. Since the reflection mechanism is deterministic, it is possible to compute the position of the aircraft whose signal was responsible for the target provided the reflecting surface parameters are known.

The geometrical situation that exists when a false target is produced is depicted in Figure 3-5. The distance d to the reflector, azimuth extent ψ_S to ψ_E of the surface, and orientation angle ϕ are assumed to be specified parameters. Any target report not correlated to a real track whose azimuth falls within the extent of the reflector is checked to determine whether it is false. First the range ρ' and azimuth θ' of the postulated real aircraft are computed. Then the system tracks are searched to see whether any are near that location. If one is found that agrees on code and altitude with the suspect report, the report is labelled false.

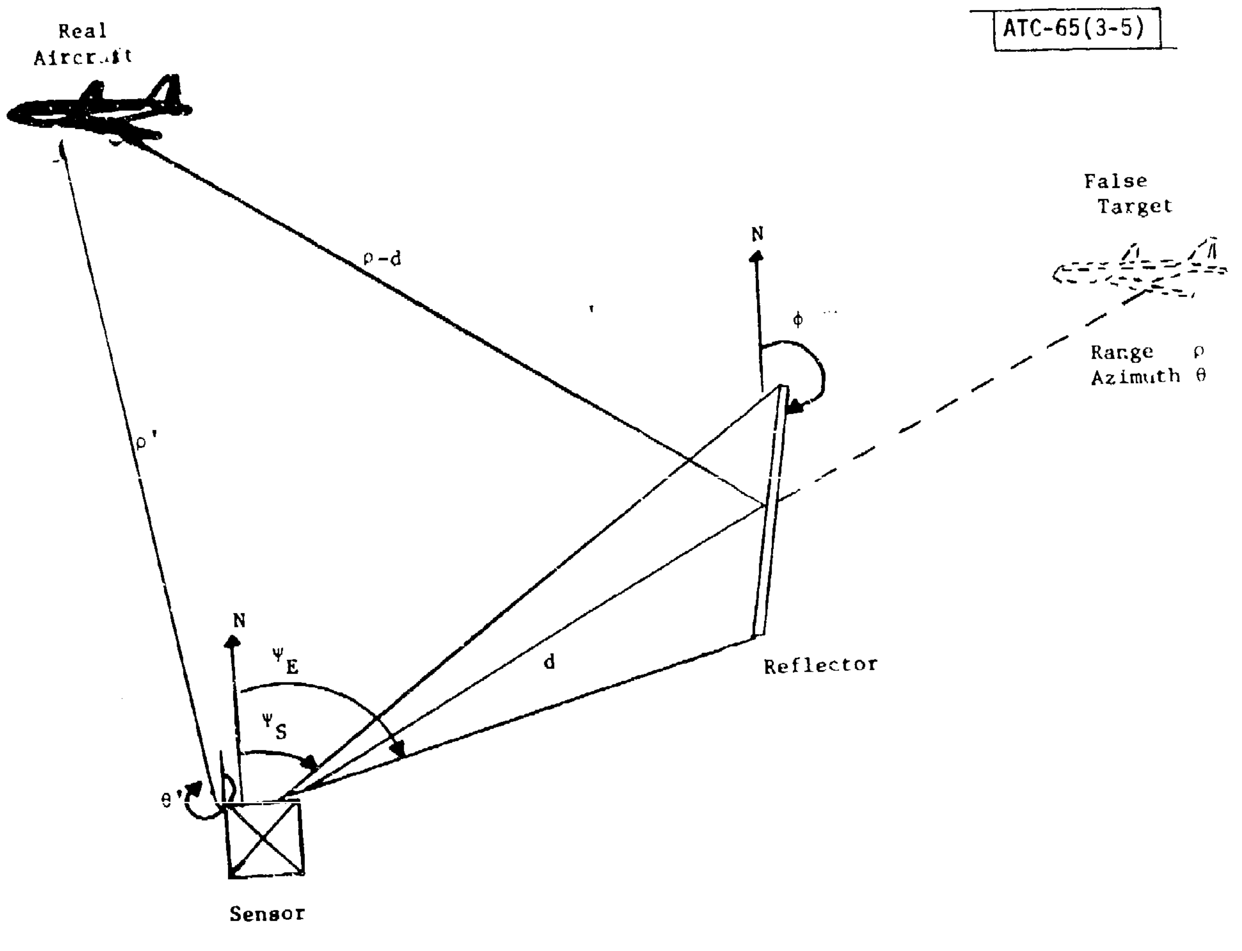


Figure 3-5: False Target Geometry

If a target report is formed by fruit replies that coincidentally correlate with each other, the report will virtually always consist of 1 mode A reply and 1 mode C reply. This is because replies of opposite modes need not agree on code to correlate while two replies of the same mode require total code agreement. Thus, when such a 2-hit report fails to correlate with a track, it is suspected of being a fruit report. The confirmation is the absence of a reinforcing radar report, as a fruit target report will not correspond to any real aircraft.

Split reports occur when the reply sequence from an aircraft is separated by the reply correlation process into two or more target reports. This can result from code or azimuth declaration errors in the reply processor, from intermode delay variations in aircraft transponders (which result in range splits), or from various environmental effects. Many of the more common types of splits have easily recognizable characteristics that permit them to be identified. The less valid of the two reports can then be discarded.

Ringaround target reports occur when sidelobe interrogations are received successfully by an aircraft, and its replies are not rejected as sidelobe by the sensor antenna. This will generally occur when an aircraft with a faulty transponder is flying overhead. In addition, monopulse system failures at high elevation angles can also lead to ringaround. The algorithm for identifying ringaround targets is very similar to that for identifying reflection false targets. In this case, the "reflector" is the sensor itself, and all azimuths are inspected. Any high elevation angle target report not correlating with a real track is subjected to the ringaround test.

The above false alarm tests apply to discrete and non-discrete targets alike. An additional test is applied only to discrete reports to identify other forms of false alarm targets, especially those caused by ground reflections. The test is that if two reports have the same discrete code, and are close together in range and azimuth, the longer range report is flagged as false. This test is legitimate since discrete codes are almost always uniquely assigned to aircraft. Non-discrete codes, being assigned to many aircraft, could conceivably pass this test when two real aircraft existed. Thus, the test cannot be applied to them.

3.9 Primary Radar Utilization

Primary radar reports can aid the ATCRBS surveillance system in two major ways. First, such reports can improve tracking on ATCRBS equipped aircraft by reinforcing beacon reports and by filling in for missing beacon reports. Second, the radar reports will permit surveillance to be maintained on non-ATCRBS equiped aircraft. The first function will always be employed in the system, while the second is an option.

The various manners in which radar reports interact with the surveillance processing functions described in this chapter are summarized by the following sequence of events:

1. First attempt to correlate radar reports with beacon reports; those radar reports which achieve successful correlation are not processed further.

2. Then attempt to correlate remaining radar reports with coasting beacon tracks; those successfully correlating are used to update the beacon tracks and are not processed further.

3. Then attempt to correlate radar reports not used above with radar tracks; those successfully correlating are used to update these tracks.

4. Finally, use remaining radar reports to initiate new radar tracks.

Association of radar and beacon reports is based solely on geometry, as no code or altitude information exists in a radar report. All radar reports that fall within a specified range and azimuth box centered at the beacon report position associate with that report. The closest such radar report (if any exist) will then be chosen to reinforce the beacon report.

The selection of the radar report to use to update an uncorrelated beacon track is performed by exactly the same procedure as that described previously for the selection of the best beacon report, except that no code or altitude information exists. Should a radar report be chosen, it is used to update the beacon track position in exactly the same manner as if it were a beacon report.

Finally, leftover radar reports are used to update existing radar-only tracks or to initiate new ones. The radar report to radar track correlation algorithm, the radar track initiation procedure, and the radar track update mechanism are all identical to the corresponding beacon procedures. The rationale for employing parallel rules for all the radar and beacon processes is that the same program subroutines can be employed for both, thereby saving substantial memory and programming costs.

4.0 REPLY CORRELATION AND TARGET FORMATION

Each time the reply processing hardware completes a reply declaration operation, it passes to the ATCRBS software subsystem the data block shown in Figure 2-6 for the reply identified. After a sweep is completed, it is the function of the reply correlation program to correlate these replies with ones received on previous sweeps, and to declare as raw target reports those groupings which are completed. In the normal mode of operation, all groupings of two or more replies are declared as raw target reports, as well as a special subset of the uncorrelated replies (as defined below); other uncorrelated replies are rejected as fruit. All reply correlation operations should be finished before the information for the next sweep arrives if unbounded system delay is to be avoided.

As stated in Chapter 1, mode 2 replies are not treated with the same importance as mode A or C replies in this ATCRBS implementation. Whenever mode 2 replies are available to the sensor, the function of reply correlation is to associate the proper mode 2 code with each declared target report. However, these replies are not used to create a target report; the two reply minimum refered to above must be met by mode A and C replies only.

4.1 Software Reply Declaration

The first function performed by the reply correlation subsystem is the completion of the reply declaration procedure begun by the hardware reply processor. This function first searches for potentially extraneous replies that might have arisen from

1. Sidelobe interference,

2. A military identification response, or

3. An out-of-spec (wide pulse) transponder.

Any such reply that satisfies the confirmation test corresponding to its category (described below) is rejected. All remaining replies then have their actual range and azimuth computed from the time and monopulse count values supplied by the reply processor. Figure 4-1 presents a flowchart of this initial function.

By design, the transmitted signal mainbeam is wider than the received sidelobe suppression (RSLS) region. Thus, it is not unusual for sidelobe replies from an aircraft to exist on either side of the accepted mainbeam replies. Should two aircraft, somewhat offset in azimuth, be synchronously garbling each other, the set of successive sweep replies depicted in Figure 4-2 would result.

Depending upon the detailed code pulse structure and amplitudes of the two garbling replies, six different situations could exist in which the sidelobe and mainbeam replies on the end sweeps produce hybrid brackets, defined as ones in which one framing pulse is mainbeam and the other sidelobe.

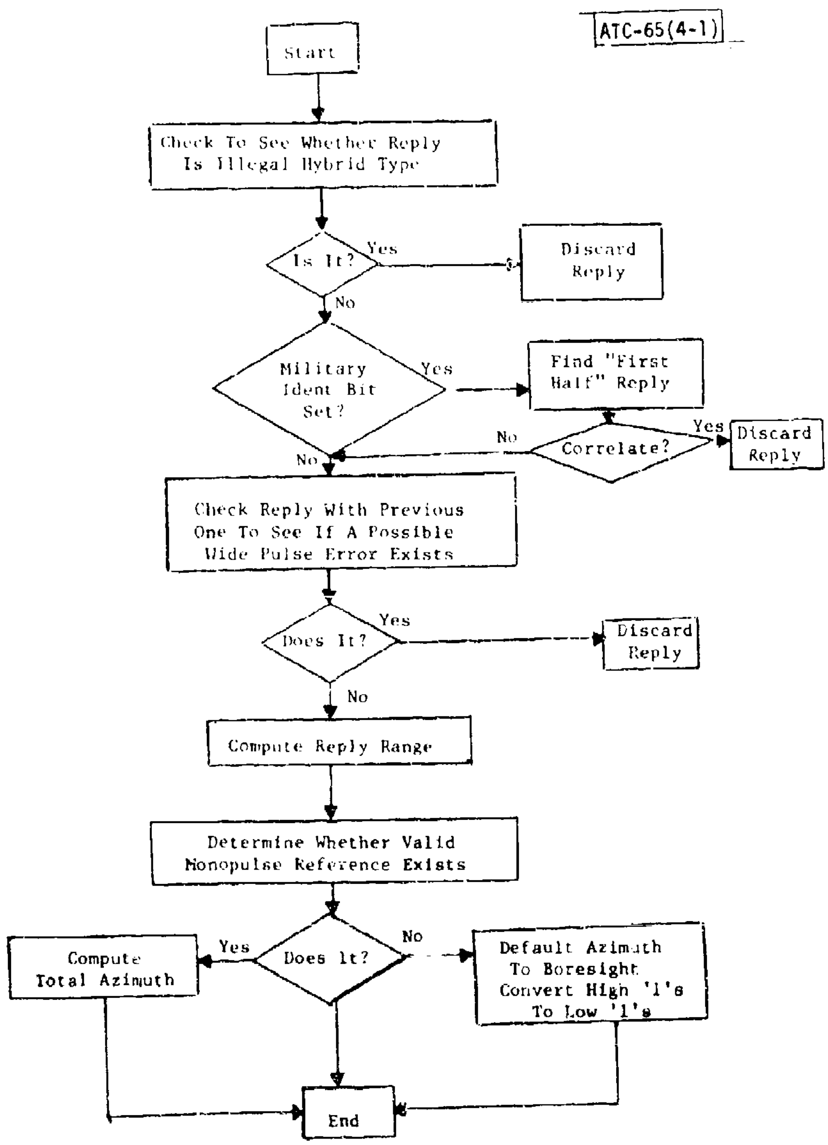
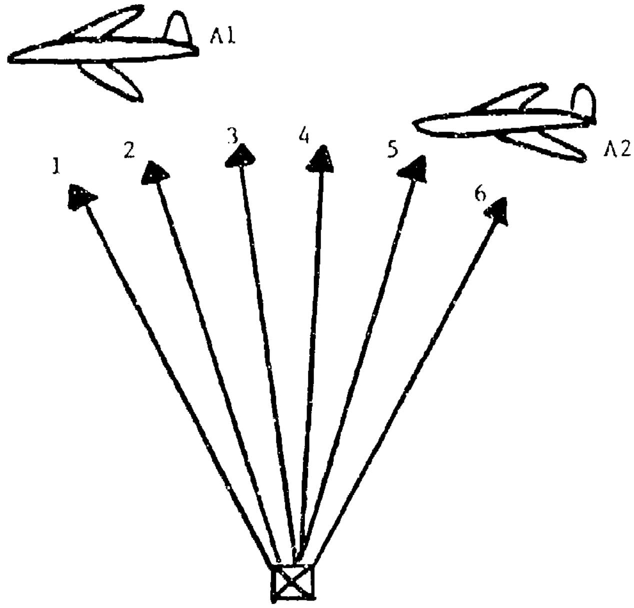
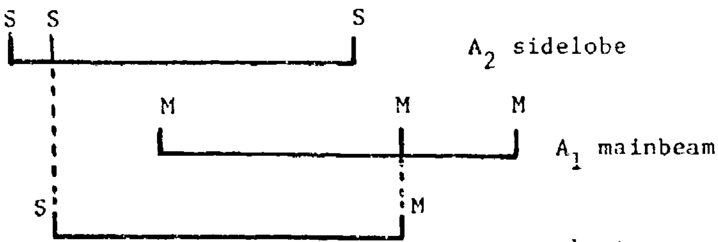


Figure 4-1: Initial Reply Correlation Function



Sweeps 1 and 2: A_1 only
Sweeps 5 and 6: A_2 only
Sweeps 3 and 4:



phantom produced, and kept since
A_2 bracket is rejected as sidelobe

Figure 4-2 : The Phantom Problem

These cases, and the replies that would be declared by the reply processor, are shown in Figure 4-3. The reply processor logic accepts all hybrid brackets, but discards purely sidelobe ones. Thus, the phantoms of cases 1 and 2 could not be identified as such.

In cases 3 and 5, the hybrid bracket represents a valid reply; these situations account for the acceptance of hybrid replies. Unfortunately, the hybrid replies in all other cases are extraneous replies that should be discarded. The method that can be used to distinguish the two valid cases from all the others is really quite simple. A study of the six cases proves the validity of the following rule:

> a reply, each of whose framing pulses is either sidelobe or garbled, should be discarded.

In other words, all valid replies must contain at least one ungarbled mainbeam framing pulse. The reply processor output for a reply, as noted in Figure 2-6, specifies which framing pulse (F1 or F2) was used as the monopulse reference, and whether this pulse was mainbeam or sidelobe. These pieces of information suffice to allow implementation of the rule.

The following facts can all be gleaned from Chapter 2:

1. The F2 pulse is used as the reference if and only if F1 was garbled.

2. If the F2 pulse is the reference, it must be ungarbled (otherwise, by fact 1, the reply would have been eliminated as a phantom).

3. If the F1 pulse is the reference, and it is labelled sidelobe, the F2 pulse must be mainbeam (since replies with both brackets sidelobe are not declared).

Thus, the resolution procedure for the four possible cases becomes:

1. F1 reference, mainbeam: accept the reply, as F1 is ungarbled and mainbeam

2. F1 reference, sidelobe: accept if F2 is ungarbled, reject other wise (see below)

3. F2 reference, mainbeam: accept the reply, as F2 is ungarbled and mainbeam

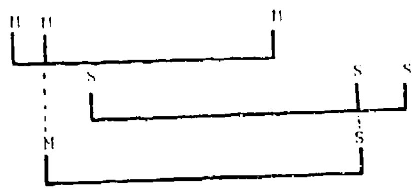
4. F2 reference, sidelobe: reject the reply, as F1 is garbled and F2 is sidelobe

The only method that can be employed in case 2 to determine whether F2 is garbled is to check each subsequent reply j to see whether any satisfy the garble condition relative to the suspect reply i:

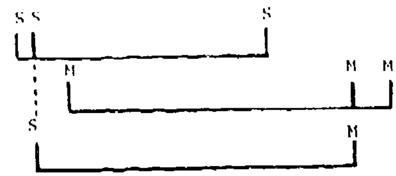
$$time_j - time_i = 24N \pm 2 \qquad N = 1, 14 \quad (16.552\text{ MHz clock})$$

Since replies are range ordered, once a reply j is reached that exceeds reply i by 339 counts, the test is concluded.

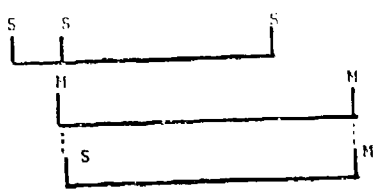
ATC-65(4-3)



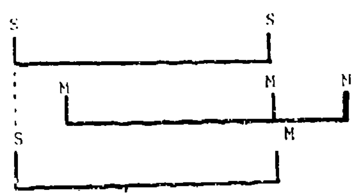
Phantom accepted, mainbeam kept.
Case 1



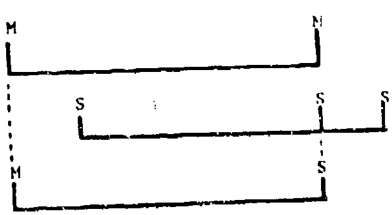
Phantom accepted, mainbeam kept. Case 2



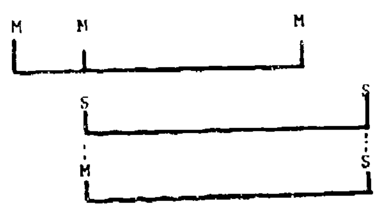
Mainbeam becomes hybrid, side-lobe rejected.
Case 3



Sidelobe becomes hybrid, mainbeam kept.
Case 4



Mainbeam becomes hybrid, sidelobe rejected.
Case 5



Sidelobe becomes hybrid, mainbeam kept.
Case 6

Figure 4-3 : Producing Hybrid Replies

As explained in the previous chapter, no reply thought to be the second half of a military identification response can be discarded by the reply processor. Instead, the military bit is set in the information block for such a reply. The reply correlation software must then examine every such reply to determine whether it should be rejected. First, the earlier reply of the pair must be located. Since the suspect reply's F1 pulse falls in the SPI position of the reply sought, the relationship between them is:

$$time_2 - time_1 = 408 \pm 2 \qquad \text{(16.552 MHz clock)}$$

Replies forming a military identification pair should agree on azimuth, while two unrelated replies coincidentally satisfying this range condition would usually fail to correlate. Thus, if

$$|az_2 - az_1| \leq 10 \text{ monopulse counts} \qquad (\text{about } 0.25^{\circ})$$

the suspect second reply is discarded.

The final source of extraneous replies is an aircraft with a transponder that generates wide, out-of-spec, pulses. The reply processor will decide that such wide pulses are caused by overlapping pulses from two different replies. The result of such an error is the creation of two replies very close in range, one using the first pulse of each supposed overlapped pair and the other the second one. Since both replies are due to the same aircraft, the azimuth and code should be the same for both. Thus, any reply i is eliminated that agrees as follows with the previously received reply:

1. $time_i - time_{i-1} \leq 10$ counts
2. $|az_i - az_{i-1}| \leq 10$ monopulse counts
3. $code_i = code_{i-1}$

Replies that survive the above tests have their true range and azimuth values computed. The range of a reply is determined as:

$$range = (time - k_{offset}) * k_{convert}$$

The constant k_{offset}, which may be different for each sweep mode, is the time that would be reported for a zero range reply. Transponder and reply processor delays enter into this number. The factor $k_{convert}$ is the conversion constant between time counts and range units, and depends upon the hardware clock frequency and the value of the least significant range bit.

The azimuth of a reply is given by the boresight value of the antenna at reply reception plus the off-boresight monopulse correction. The

former value is provided directly in each reply data block, while the latter can be calculated via a table lookup from the final monopulse reference value of the reply. There are four instances in which no valid monopulse azimuth will exist:

1. No monopulse reference could be generated

2. The monopulse reference was never confirmed by a correlating pulse

3. The monopulse reference is outside the usable region

4. The monopulse reference pulse was labelled as sidelobe

The first case can occur due to a variety of wide pulse phenomenon, and is signalled by a zero monopulse reference value. The second situation could be caused by heavy garble, or by an incorrect initial reference value; it is flagged by the "$N \geq 2$" bit of the reply data block being set to zero. The third case arises whenever the reply was received sufficiently far off-boresite to be outside the calibration region of the monopulse antenna. Finally, if the monopulse reference was initialized by a sidelobe pulse, as indicated by the corresponding reply bit, it is highly suspect and thus not used.

Whenever none of these special cases are present, the reply azimuth is given by:

$$\theta = \psi_{bs} + T(\psi_{ref})$$

where T is the monopulse calibration table. If this monopulse correction is ≥ 0, the reply is labelled as a side 1 reply, else it is called a side 2 reply. Should one of the special cases apply to the reply under consideration, however, the reply azimuth can only be defaulted to boresight:

$$\theta = \psi_{bs}$$

and the reply specially flagged as side 0. In addition, any code bit of such a reply labelled as a high confidence '1' must be changed to a low confidence '1', as the azimuth correlation decision required for high confidence cannot be trusted.

The data structure created for each valid reply as a result of the reply declaration function is presented in Figure 4-4.

4.2 Reply Correlation Data Structures

The two key data structures employed in the reply correlation process are the reply buffer and the reply sort table. The reply buffer is a cyclic file that contains entries for all replies received on at least the last S

ATC-65(4-4)

<table>
<tr><th>0 … 7</th><th>8 … 15</th><th>16 … 23</th><th>24 … 31</th></tr>
<tr><td colspan="2">Range</td><td colspan="2">Azimuth</td></tr>
<tr><td colspan="2">Code</td><td colspan="2">Confidence</td></tr>
<tr><td>-0-</td><td>Azimuth Side</td><td>-0-</td><td>Mode</td></tr>
<tr><td colspan="2">-0-</td><td colspan="2">-0-</td></tr>
</table>

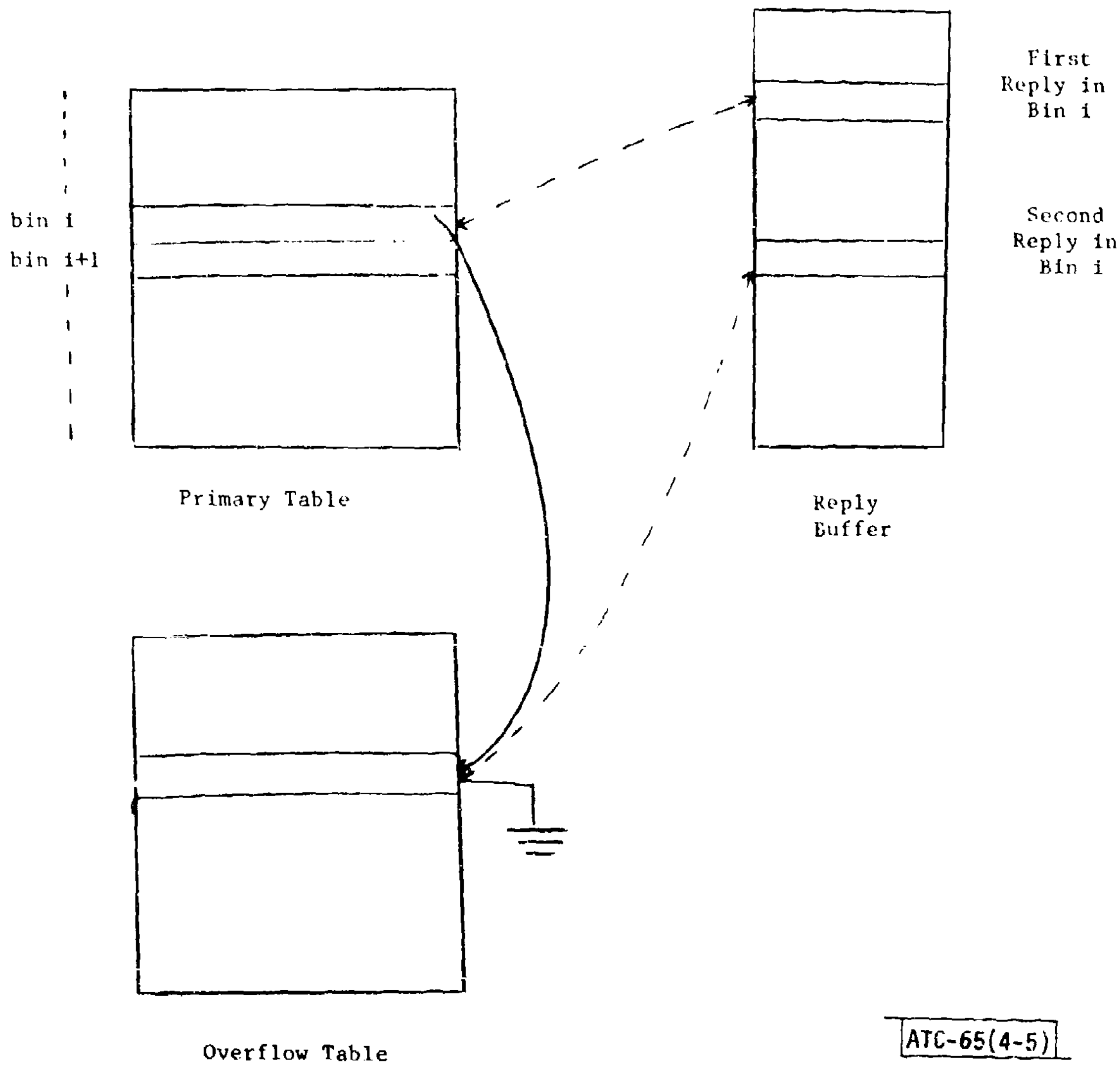
Figure 4-4: Initial Reply Data Block

sweeps, where S is a parameter (usually set at twice the expected aircraft runlength). Entries in this buffer progress from raw reply entries to completed target report entries as the reply correlation procedure advances. This multiple utilization approach minimizes data transfers as well as storage. The reply sort table permits access to replies by range quanta, and thus greatly accelerates the reply correlation operation.

The reply sort table consists of a number of bins, one for each range quantum in the sensor coverage field. The size in range units of the quantum represented by each bin is an integer power of two, which permits the bin for each reply to be computed simply by a shift and add one procedure. The memory implementation chosen for this sort table is illustrated by Figure 4-5. The primary table has one word for each bin, this word used to reference the first reply grouping in the corresponding range quantum. Additional entries in the bin are then placed in the first available slot of the overflow area. These entries are located by traversing the pointer path that begins in the primary word for the bin.

The only required fields in a sort table entry are a pointer to the reply group being represented and a pointer to the next bin entry (if any exists). However, considerable time is saved in the reply correlation process by including the most important reply attributes in the bin word, so that most noncorrelation conditions between a candidate reply and the represented group can be determined without having to access the information in the reply buffer. The two attributes chosen, as shown in Figure 4-5, are the low order range bits of the first reply in the group and the number of the interrogation sweep for the last reply in the group. The former item provides a finer test of range compatibility than just residence in the same bin, while the latter item will exclude from consideration all groups that have already experienced correlations on the current sweep (two replies on one sweep from one aircraft being impossible). The sweep number is stored on a modulo S basis, as only S sweeps are active at any instant of time.

The initial format of a reply entry in the reply buffer was shown by Figure 4-4. After a reply is processed, this format is altered in a manner dependent upon the result of the reply correlation process. The set of possible new formats is presented in Figure 4-6. If a candidate reply fails to correlate with any existing reply group, or correlates only with mode 2 replies, the minor format change shown in Figure 4-6a is affected. If a candidate mode A or C reply successfully correlates with a previously uncorrelated mode A or C reply, the entries for <u>both</u> of these replies are altered considerably. The entry for the old reply (Figure 4-6b) now includes all the attributes required for the reply correlation tests (range, azimuth, and codes and confidence words for both modes A and C), while that for the new reply (Figure 4-6c) is used to store the additional items of information required during target formation. The former entry is accessed via the sort table pointer, the latter by a pointer in the first entry. Finally, if a candidate reply of any mode successfully correlates with a reply group having two or more



0 – 7	8 – 15	16 – 23	24 – 31
Reply Number	Low Order Range Bits	Sweep Number	Next Bin Entry Pointer

Sort Table Entry

Figure 4-5: Reply Sort Table

0 7,8 15 16 23,24 31

Range	Azimuth
Code	Confidence

Mode 2 Pointer	Azimuth Side	No. of Mode 2 Replies	Mode

Sort Table Pointer	-0-

(a) Uncorrelated Reply (or one Correlated only to Mode 2 Replies)

0 15 16 31

Range					Azimuth				
Mode A Code	1	1	X	SPI	Mode A Confidence	0	0	X cf	SPI cf
Mode C Code	0	0	0	0	Mode C Confidence	0	0	0	0
Sort Table Pointer					Pointer to 2nd Reply				

(b) First Mode A or C Reply Of A Group

0 7,8 15 16 23,24 31

Σ Weighted Ranges			
Σ Weighted Azimuths			
Weighting Factor	No. of Mode 2 Replies	No. of Mode A Replies	No. of Mode C Replies
Mode 2 Code		Mode 2 Confidence	

(c) Second Mode A or C Reply of A Group

ATC-65(4-6)

Figure 4-6: Reply Buffer Formats

mode A or C replies, its entry is not changed at all; instead, the information it supplies is used to modify the first and second reply entries for the group.

4.3 Reply Correlation Concepts

The basic objectives of the reply correlation process are to associate successive sweep replies from the same aircraft and to eliminate all fruit replies. In addition, however, since the program developed for this purpose must run in real time in a real computer, the algorithms that implement these functions should execute in minimum time while requiring minimum storage. Clearly, a performance tradeoff must exist.

The major features of this reply correlation implementation can be summarized as follows:

1. Sweep-to-sweep correlation is performed during the time the replies for a new sweep are sorted, thereby eliminating both a two pass operation and the need for any association tables.

2. All replies received from an aircraft are used in target declaration; neither the first nor the one received after a miss are eliminated by the software defruiting mechanism.

3. Fruit replies are automatically eliminated from the system without need for special defruiting logic; that is, no fruit declaration mechanism is required.

4. The <u>first</u>, rather than the <u>best</u>, possible correlation is accepted for a candidate reply; at the cost of making an occasional correctable error, this rule shortens the search time and eliminates the need for complex decision logic.

5. Replies with uncertain codes (due to synchronous garble or interference) are considered for association after those with high confidence codes, thus minimizing ambiguous situations and cross correlations.

6. A new reply is correlated with an existing reply group only if it (a) falls within a specified range and azimuth box centered at the last reply in the group, and (b) agrees in all high confidence bits with the code of the same mode for the group (one bit difference is permitted for mode C replies to account for altitude level changes). Part (b) is waived for mode 2 replies.

7. Uncorrelated mode A or C replies that satisfy the range and azimuth conditions of item 6 with any reply group, but fail on the code test, are declared as 1-hit targets to permit later correction of high confidence bit errors.

Detailed descriptions of the algorithms accomplishing these features will be described below, but first a few general comments clarifying these statements will be made.

The reply entries in the reply buffer are processed in two separate passes: the first for reply correlation and the second for target declaration. All replies that fail to correlate on the correlation pass are sorted. They will then be available for correlation with future replies from the same aircraft when those replies are found. Thus, no replies are lost, and holes in the reply sequence are unimportant. Except in low fruit environments when 1-hit reports are permitted, the target declaration pass searches for reply entries that have a non-zero correlation pointer field (see Figure 4-6) and creates a target report for each such reply found; other replies are ignored. Thus fruit replies, which have never correlated, are automatically passed over by the program and become discarded without any formal declaration.

An ambiguity will arise in the reply correlation process whenever two mode A or C replies in the same sweep could correlate with the same existing reply grouping, or one mode A or C reply could correlate with two such groupings, or both. The selected resolution method, choosing the first possible correlation, will occasionally produce errors when these ambiguities arise. However, the approach chosen has two major advantages over any other that could be devised. First, the overwhelming majority of all replies could only correlate with one existing grouping and vice versa. The first choice method ends when this match is identified, whereas any other approach would have to look for all other possible matches. Second, although the first choice logic is far simpler to execute than any other, it has been found to most often make the correct decision.

The categories of errors that could occur in an ambiguity resolution are the following:

1. Two replies (or groupings) exist with the same code, and the wrong one is selected.

2. Two new replies with different codes compete for an existing reply grouping that has not yet established a code for their mode, and the wrong code is selected; this situation could occur as a result of:

   (a) two aircraft crossing

   (b) a fruit reply at the same location as the real reply

3. Due to the existence of low confidence code bits, two replies with different codes are both able to match the code of one of the existing reply groupings, and the incorrect choice is made.

The first type of error will at worst produce a target report with a small range and azimuth error. Since the reply correlation window is much smaller than the target-to-track correlation one, the error will never be critical. The second type of error will produce one or more target reports with the wrong code/altitude pairing. This mistaken mode pairing will become obvious during target to track correlation, as the track file will contain the proper pairing. Then, since all the replies involved in the ambiguity situation were maintained, even if some failed to correlate (refer to rule 7 above), the "code swapping" mechanism built into target to track correlation (described in Section 6.4) will be able to undo this error and construct the proper pairings. Finally, the third type of error is really a common subset of the second. To reduce the possibility of such errors, replies with all high confidence code bits are processed first by rule 5.

The correlation requirement of exact code agreement (or one bit difference for mode C) between a new reply and an existing reply group was chosen to maximize the code information maintained in the system. If a number of code bit differences were permitted in correlation, the corresponding positions would have to be set to low confidence. Should, in fact, a high confidence code error occur in a reply, rule 7. guarantees that the alternate code (i.e.: the one not contained in the target report) is available. Then the code swapping mechanism mentioned above will be able to place the proper code in the target report.

On the other hand, no mode 2 code is maintained in a track file; thus, no mode 2 swapping is possible. For this reason, and to prevent many-way target splits, mode 2 code agreement is not required for reply correlation. Similarly, all uncorrelated mode 2 replies are discarded as fruit.

4.4 Reply Correlation Rules

The reply correlation algorithm outlined above is a correlate-while-sort process. Each reply, in turn, is examined to determine its proper sort table range bin. It is then compared sequentially with each reply group (single reply or correlated group) represented by that bin. The first such group is represented by the entry in the primary word for that bin, while the others are located by following the pointer chain emanating from that word (see Figure 4-5). The reply will be correlated with the first group that satisfies all the matching criteria and be added to it in the manner described below. If no match is located, but the range of the reply is sufficiently close to a bin boundary to permit correlation with a group in the adjacent bin, that second bin is searched. If still no matching group is found, a new sort table entry is created for the reply in the original range bin.

A new reply will correlate with an existing group of replies only if all four of the following conditions are met:

1. The range difference between the reply and the group is no greater than $\Delta\rho_{max}$.

2. The monopulse azimuth difference between the reply and the group is no greater than $\Delta\theta_{max}$. If one of the azimuths is defaulted to boresite, this test is bypassed.

3. The group has not already been correlated with another reply from the current sweep.

4. The code of the reply (identity code or altitude) is compatible with the corresponding code of the group. If the group doesn't yet possess a code of the corresponding mode, this test is automatically satisfied. This test is waived for mode 2 replies.

The range test is simplified by the presence of the low order bits of the reply group range in the reply sort table entry. The test thus becomes:

$$|(\rho_{reply} - [bin\ \# - 1] * 2^B) - \text{low order range}| \leq \Delta\rho_{max}$$

$$2^B = \text{sort bin quantum}$$

and no reference to the reply buffer is required for this primary test. The azimuth comparison cannot be trusted if either the candidate reply or reply group azimuth is boresite, as the possible error in such an estimate is equal to the beamwidth. Hence, in either case the test is bypassed. A reply (or uncorrelated reply group) azimuth is boresite if the boresite indicator (see Figure 4-6a) is set, while a correlated reply group azimuth is boresite if the weight field of the second entry (see Figure 4-6c) is less than 32. If both azimuths are monopulse, the test is:

$$|\theta_{reply} - \theta_{group}| \leq \Delta\theta_{max}$$

If a candidate reply passes both the range and azimuth tests with respect to any existing group, the code swap Boolean variables associated with both that reply and the first reply of the existing group will be set to TRUE if the correlation is blocked by failure of either remaining test. This action will insure that both replies, if mode A or C, are declared as target reports whether or not either becomes correlated, and that both will be available for possible code swapping later.

The third test, prior correlation of the group, is performed by comparing the number of the current sweep (modulo the runlength) with the sweep number field in the reply sort table entry of the group. If these values are equal, the group has already correlated with another reply on the current sweep, and further correlation is forbidden.

A mode A (identity) reply and a reply group with has mode A code established are defined as being compatible when all of their common high confidence bits agree. Mathematically, this condition can be expressed as follows:

$$|(\overline{A \oplus C}) \vee B \vee D| = 12 \qquad \text{(i.e.: all bits of result are '1')}$$

where A, B are reply code and code confidence words
C, D are group mode A code and code confidence words

Figure 4-7 demonstrates sample agreement and disagreement situations. Implicit in this test is that the probability of an aircraft changing its identity code during the runlength of a scan (nominally .03 seconds) is nearly zero.

On the other hand, an aircraft is reasonably likely to change its altitude level during this period of time, although a change of two levels is impossible. Thus, utilizing the fact that altitude encoding employs a Gray code, a mode C reply and a group with an established altitude are defined as compatible when they disagree on at most one mutually high confident bit, or:

$$|\overline{(A \oplus E)} \vee B \vee F| \geq 11$$

where E,F are group mode C code and code confidence words

This test is clearly necessary, given the properties of a Gray code, but not sufficient, as the bit that differs may not be the one that represents a single level change. However, the test has been deemed adequate as it will reject most incorrect reply correlations, and as determining the bit that should change is very time consuming. Examples of how this test is utilized, including an incorrect acceptance situation, are given by Figure 4-8. It might be noted that on most computers, the magnitude test can be implemented considerably more efficiently than the twelve shifts and twelve comparisons it would appear to require.

An alternative way to have determined altitude compatibility would have been to convert both altitude codes into flight levels and then to have made a simple subtraction. The two would then be compatible provided that:

$$|(\text{reply flight level}) - (\text{group flight level})| \leq 1$$

This test, although more accurate as well as simpler than the one presented above, assumes that the flight levels are known with certainty. Unfortunately, the possibility of having low confidence encoded bits cannot easily be included into this test, as one uncertain Gray code bit can translate into many uncertain binary bits. Thus, such a test can degenerate to automatic acceptance when interference is present. Furthermore, the requirement of decoding every mode C reply, including fruit, rather than just decoding target reports, would place a large processing burden on the system.

If at any time during the search through the linked list of bin entries a null entry is found, that is, one which no longer represents a group of replies, the list is patched around that entry. Thus, subsequent searches through the bin will be shorter and quicker. A null entry will arise whenever an old group of replies is expunged from the system. Since no backward pointers are contained in the bin entries, it is impossible at that time to remove the entry.

ATC-65(4-7)

Reply Code (A):	001	010	001	010
	X	XX	XX	X
Group Code (C):	011	100	010	011
Reply Conf. (B):	010	101	010	101
Group Conf. (D):	000	011	001	000
$\overline{(A \oplus C)}VBVD$:	111	111	111	111

<u>agreement</u>

Reply Code (A):	100	110	000	010
	XX	X	XX	XX
Group Code (C):	111	010	011	001
Reply Conf. (B):	010	001	001	001
Group Conf. (D):	000	100	100	000
$\overline{(A \oplus C)}VBVD$:	110	111	101	101
	*		*	*

<u>disagreement</u>

X - possible mismatch
* - actual mismatch

<u>Figure 4-7:</u> Mode A Agreement Testing

ATC-65(4-8)

Assume all confidence bits are high confidence (i.e.: B and F are all '0's.)

		Flight Level
Reply Code (A):	010 011 001 000	123
Group Code (E):	<u>010 111 001 000</u>	122
$(\overline{A \oplus E})VBVF$:	111 011 111 111	
	weight = 11, <u>acceptance</u>	

		Flight Level
Reply Code (A):	110 100 110 000	76
Group Code (E):	<u>010 000 001 000</u>	143
$(\overline{A \oplus E})VBVF$:	011 011 000 111	
	weight = 7, <u>non-acceptance</u>	

		Flight Level
Reply Code (A):	101 010 100 000	247
Group Code (E):	<u>100 010 100 000</u>	48
$(\overline{A \oplus E})VBVF$:	110 111 111 111	
	weight = 11, <u>acceptance</u>, <u>error</u>	

<u>Figure 4-8:</u> Mode C Agreement Testing

The result of this reply correlation procedure is that the candidate reply could fail to correlate with any group, correlate with a previously uncorrelated reply, or correlate with an existing group of replies. If the first case applies, the only action taken is the creation of a reply sort table entry for the reply. Thus, fruit reply processing is neglibible. The actions performed for the correlation cases are covered in the next sections.

A flow chart of the reply correlation algorithm described here is presented in Figure 4-9.

4.5 Reply Group Updating for Mode A and C Replies

When a previously uncorrelated mode A or C reply is joined by a new mode A or C reply, the two reply entries in the reply buffer must be altered to conform to the format previously defined in Figures 4-6b and 4-6c. The first step in this transformation is to use the parameters of the old reply (whose format is given by Figure 4-6a) to construct the initial information blocks shown in Figure 4-10. This figure assumes the earlier reply was of mode C; a mode A reply would have been handled in the analogous dual manner. Note that the undefined mode (in this case mode A) is set to the default condition, all bits low confidence 1's. If the earlier reply had previously correlated with one or move mode 2 replies, as indicated by a non-zero mode 2 pointer, the mode 2 information for the second block is taken from the referenced mode 2 reply (refer to the next section for a discussion of mode 2). Otherwise, the mode 2 code is also defaulted.

The weight field in the second data block indicates on which side of boresite the reply was received. This is important because the target range and azimuth estimates are defined to be the average of the two replies nearest to boresite, one on either side. If replies are received on only one side of boresite, the range and azimuth of the single reply nearest to boresite will be used. The weight encoding that has been adopted for this first reply is:

weight = 1 : boresite reply (no monopulse estimate)

weight = 32: side 1 reply (monopulse correction $\geq$ 0)

weight = 64: side 2 reply (monopulse correction < 0)

The range and azimuth of boresite replies will not be employed in target declaration if any monopulse samples exist. If all replies are defaulted to boresite, however, a simple beamsplitting averaging method will be employed.

Once the transformed data blocks exist, either by having been just constructed or by having been created during a previous sweep's processing, the attributes of the correlating reply (from the current sweep) are added into the structure. First, the number of hits for the mode of the new reply is incremented by one. Then the code and confidence word estimates for that mode are improved by incorporating the information from the new reply. The Boolean update equations for mode A are:

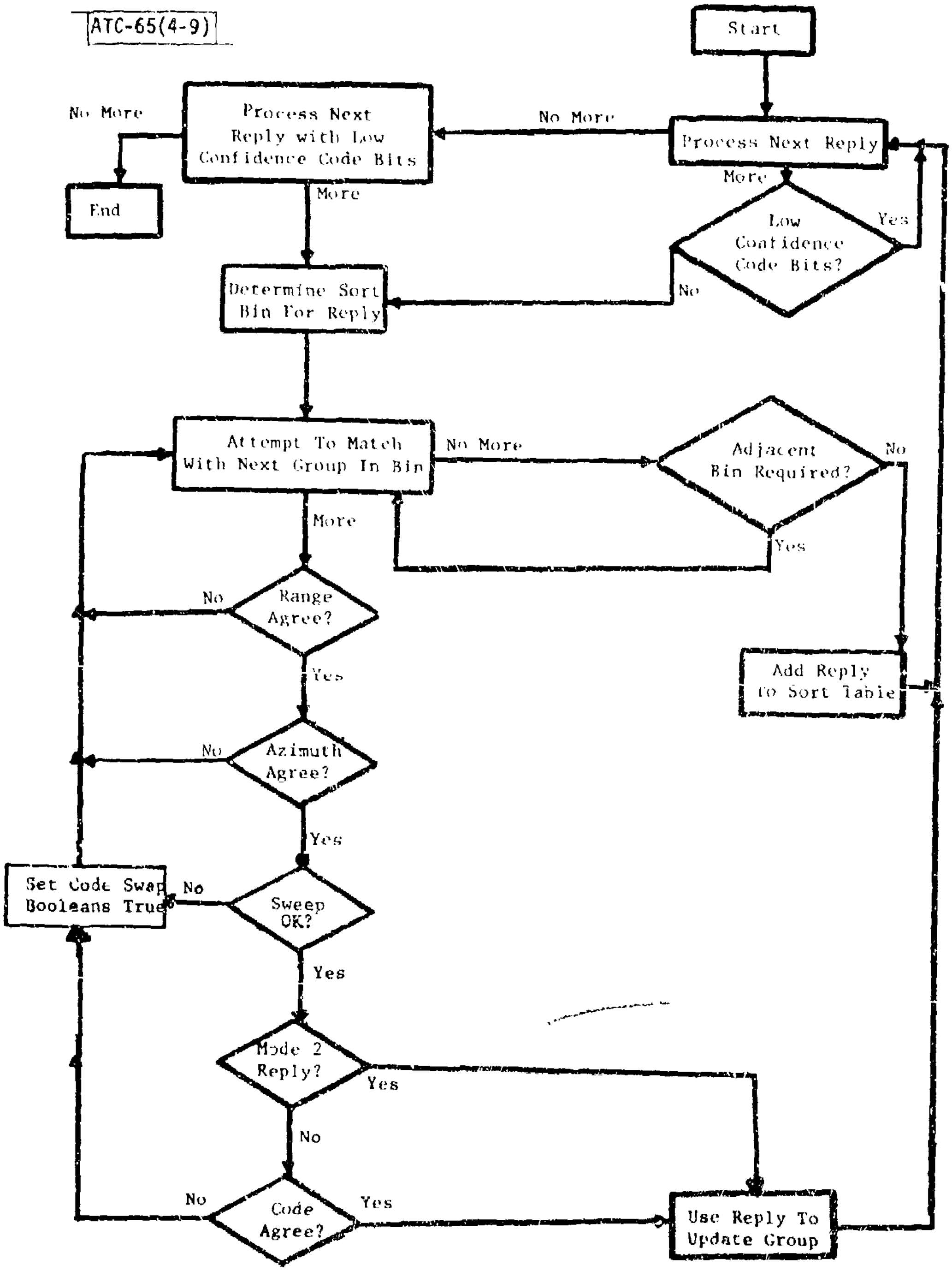


Figure 4-9: Reply Correlation Function

ATC-65(4-10)

0 — 15	16 — 31
PR Range	PR Azimuth
All '1's	All '1's
PR Code	PR Confidence
Sort Table Pointer	Pointer to New Reply

Previous Reply Buffer Entry
PR = Previous Reply

PR Range X PR Weight			
PR Azimuth X PR Weight			
PR Weight	No. of Mode 2 Replies	0	1
Mode 2 Code		Mode 2 Confidence	

New Reply Buffer Entry

PR = Previous Reply

Figure 4-10: Initial Buffer Entries for Reply Group

$$C \leftarrow A \cdot C + A \cdot \bar{B} + C \cdot \bar{D}$$

$$D \leftarrow B \cdot D$$

where A, B are the code and code confidence words of the new reply
C, D are the existing mode A code and code confidence words

These equations implement the rules shown graphically by the Karnaugh map of Figure 4-11a. Basically, the resulting bit is high confidence if either estimate of it is high confidence, and a low confidence '0' takes precedence over a low confidence '1'. Note that high confidence bit disagreement is not permitted by the reply correlation rules. The Boolean update equations for mode C, which implement the rules of Figure 4-11b, are given by:

$$E \leftarrow A \cdot E + A \cdot \bar{B} + E \cdot \bar{F}$$

$$F \leftarrow B \cdot F + A \cdot \bar{B} \cdot \bar{E} \cdot \bar{F} + \bar{A} \cdot \bar{B} \cdot E \cdot \bar{F}$$

where E, F are the existing mode C code and code confidence words.

The added complication arises because high confidence bit disagreement is permitted for mode C replies. When it occurs for a given bit position, that bit is set to low confidence '1'.

The estimates for both the X and SPI bits are updated when a mode A reply is received (these bits not being meaningful on mode C). The initial setting for each is low confidence '0'. The update equations for either bit are then identical to those for mode C presented above. Again, the equations set either estimate to low confidence '1' whenever a high confidence disagreement is found.

The updates required for the weighting factor and the weighted sum of range and azimuth words of the second data block depend upon the present state of the weighting factor and the sign of the monopulse azimuth correction possessed by the new reply. First this correction is used to determine the weight associated with the new reply as described earlier. Then the rules presented in Figure 4-12 for updating the entries in the data structure are applied. These rules implement the following ideas, all based on the assumption that successive reply monopulse corrections for an aircraft are monotonically decreasing as shown in the figure.

1. If the weighting factor already equals 64, the replies to be averaged have already been received.

2 If the weighting factor is 32 and a side 2 reply is received, the new reply is the second of the two replies to be averaged.

ATC-65(4-11)

New Reply

Group	HO	LO	L1	H1
HO	HO	HO	HO	X
LO	HO	LO	LO	H1
L1	HO	LO	L1	H1
H1	X	H1	H1	H1

H = high confidence = 0

L = low confidence = 1

X = cannot occur

Entry gives new code for group

(a) Mode A

New Reply

Group	HO	LO	L1	H1
HO	HO	HO	HO	L1
LO	HO	LO	LO	H1
L1	HO	LO	L1	H1
H1	L1	H1	H1	H1

(b) Mode C

Figure 4-11: Code Update Rules

Old Weight	Reply Weight	New Weight	New Weighted Sum of Ranges Word
< 32	1	old + 1	old + reply range
< 32	32	32	32 * reply range
< 32	64	64	64 * reply range
32	1	old	old
32	32	32	32 * reply range
32	64	64	old + 32 * reply range
64	1	old	old
64	32	old	old
64	64	old	old

Old means previous value

Weighted sum of azimuths word follows same rules as weighted sum of ranges word.

Figure 4-12: Weight Update Rules

3. If a side 1 reply is received, it must be closer to boresite than any previous reply, and thus any previous reply's contribution should be deleted.

4. A boresite reply is not needed if any monopulse reply has already been received (that is, weighting factor is 32 or 64).

5. A new boresite reply is averaged with previous replies if they were all boresite (that is, weighting factor is less than 32).

One final change must be made in these data blocks if the new reply has a monopulse azimuth: the placing of this azimuth value in the azimuth entry in the first data block. This insures that the azimuth test for the next reply correlation attempt will employ as a reference the most recent monopulse estimate. Clearly, if the monopulse antenna were perfect, any reply's azimuth would serve this purpose. However, various effects, such as frequency offset and elevation angle, often lead to a slope in the monopulse correction function. In such cases, the most recent azimuth estimate will be the best prediction of the next reply's value.

In addition to these updates in the reply buffer, an update to the reply sort table entry is required whenever a reply correlation is attained. The required action is the setting of the sweep number field to the number of the current sweep. This action prevents the group from correlating with another reply on the current sweep.

A flow chart of the reply group updating functions is presented in Figure 4-13.

4.6 Reply Group Updating for Mode 2 Replies

When a mode 2 reply correlates with another reply, the update procedure is considerably simpler than that described in the previous section. This is because the only effect a mode 2 reply can have is to improve the mode 2 code estimate connected with a target report. It cannot be used to turn on uncorrelated reply into a multiple hit reply group, nor can it be used to improve the range or azimuth estimates of an existing group.

Should a new mode 2 reply correlate with a previously received mode 2 reply, the number of replies field of the previous reply (see Figure 4-6a) is incremented by one, while the code and confidence fields of that reply are updated according to the following rules:

$$G \leftarrow A \cdot G + A \cdot \overline{B} + G \cdot \overline{H}$$

$$H \leftarrow B \cdot H + A \cdot \overline{B} \cdot \overline{G} \cdot \overline{H} + \overline{A} \cdot \overline{B} \cdot G \cdot \overline{H}$$

where A,B are the new reply code and confidence
G,H are the existing mode 2 code and confidence

These equations, which are identical to those for mode C, set a bit position to low confidence '1' whenever a high confidence disagreement is encountered.

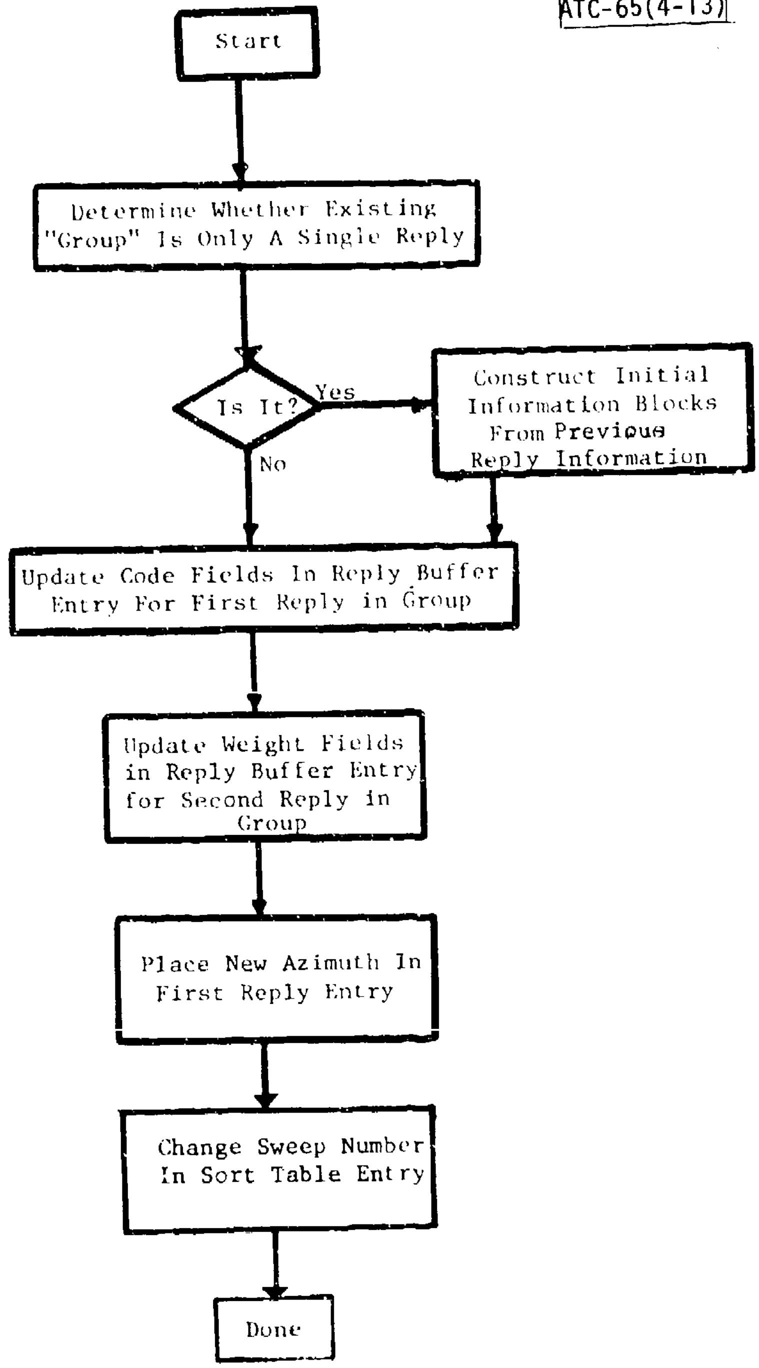


Figure 4-13: Reply Group Update Function for Mode A and C Replies

When a previous mode A or C reply is correlated for the first time by a current mode 2 reply, the earlier reply is made to point to the current reply, and its mode 2 hits field is initialized at one. Should a subsequent mode 2 reply be added to this set, this number of hits field is incremented by one and the code and confidence fields of the pointed to reply are updated by the new reply information according to the above equations.

The third possible case is that of a current mode 2 reply correlating with an existing reply grouping. In this event, the mode 2 code and confidence fields of the second reply entry (shown in Figure 4-4c) are updated by the above rules, and the number of mode 2 replies is incremented by one.

The final mode 2 correlation situation occurs when a current mode A or C reply correlates with a previous mode 2 reply. In this case, the new mode A or C reply assumes the sort table entry established for the mode 2 reply. That is, the new reply's sort field is set to that of the previous reply, the previous reply's entry is nulled, and the sort table entry itself is made to point to the new reply. Then the new reply is set to point to the previous mode 2 reply, and the number of mode 2 replies is copied from the previous reply to the current one.

The final action in any mode 2 update situation is the placing of the current sweep number into the proper field of the sort table entry (see Figure 4-5). This prevents correlation by another reply on the current sweep. Figure 4-14 summarizes in flowchart form the actions taken in each update case.

4.7 Raw Target Report Formation

After all the replies for the current sweep have been processed through reply correlation, all reply groupings begun on the oldest active sweep are declared as raw target reports. These groupings are known to be complete because the number of active sweeps was chosen to be equal to the longest possible reply runlength.

If this oldest sweep is mode 2, no target reports can be created. This is because, as stated earlier, only mode A and C replies count in determining target declarations. If a mode 2 reply correlated with such a reply, the target was assigned to the sweep of the first mode A or C reply. Thus, the target declaration process for a mode 2 sweep consists simply of removing sort entries in the manner described below.

The target declaration process for a mode A or C sweep consists of a single pass through all the reply entries. The sort table entry and correlation pointer fields of the reply (see Figure 4-6) are examined to determine the type of reply encountered. If both of these fields are null, the reply is part of a prevouisly declared target report, and hence the reply is simply

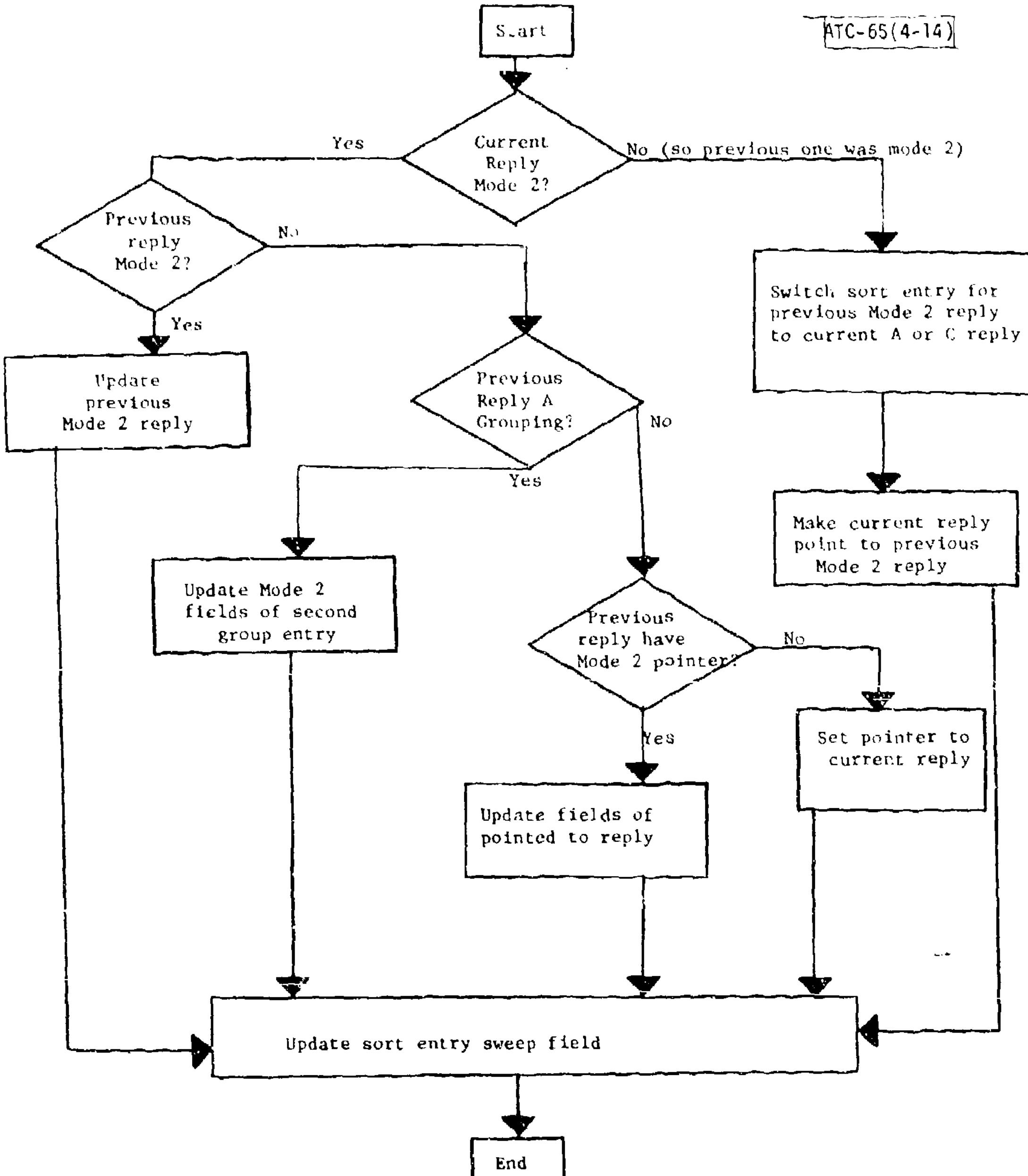


Figure 4-14: Mode 2 Reply Update Cases

passed over. If the reply has a non-zero sort table entry but a null correlation pointer, it is an uncorrelated reply. Such a reply will usually be skipped over as a fruit, but on occasion it will be needed as part of a potential code swapping situation (see sections 4.3 and 4.4). A code swap Boolean variable associated with this reply indicates which situation applies. If the variable is FALSE, the reply is not required, while if the variable has been set to TRUE (by the rules given in 4.4), a 1-hit target report is declared for this reply. However, if the fruit environment is so benign that 1-hit reports are desired, all uncorrelated replies encountered on the sweep are turned into 1-hit reports. Finally, if the reply entry has both fields non-zero, the reply is the first reply of a group, and a regular target report is created for this group.

The format employed for any target report is presented in Figure 4-15. For a 1-hit report, the values placed in the various fields are determined as follows. The range and azimuth are copied directly from the reply entry. If the reply is of mode A, the mode A code and confidence and the X and SPI bits and confidences are all obtained from the reply entry, and the number of mode A replies is set to one. The undefined mode C code and confidence are set to indicate the default condition, all bits low confidence '0', and the number of mode C replies is set to zero. On the other hand, a 1-hit mode C report will contain default mode A quantities and the mode C code and confidence as specified in the reply entry. Next, if the reply has correlated with one or more mode 2 replies, as indicated by a non-zero mode 2 pointer field (refer to Figure 4-6a), the mode 2 code and confidence are copied from the referenced reply and the number of mode 2 replies is set to the value specified in the reply. Otherwise, the mode 2 code and confidence are defaulted to all hits low confidence '0'. The correlating track number is set to zero for all raw reports, as no track correlation has yet been attempted. Finally, two of the special purpose bits apply to raw target reports. The first, the boresite target bit, is set when no monopulse azimuth exists for the reply. This condition is signalled by an azimuth side setting of zero. The other relevent special bit, potential code swap, is set if the code swap Boolean variable for the reply is TRUE.

For a multiple hit target report, all target values are determined from the information in the first two group reply entries (shown in Figures 4-6b and 4-6c). The range and azimuth of the target are calculated by dividing the respective weighted sum by the weighting factor. The modes A,C, and 2 code and respective confidence estimates, the X and SPI bits and confidences, and the number of replies of each mode, are all copied directly from the reply entries. The only change is that if any mode has no replies, its code is set to all bits '0'. The two applicable special purpose bits are determined as follows: the target is flagged as boresite if the weighting factor is less than 32, and as a potential code swap candidate if the Boolean variable associated with the first reply of the group is TRUE.

<table>
<tr><td colspan="5">0 … 15</td><td colspan="5">16 … 31</td></tr>
<tr><td colspan="5">Range</td><td colspan="5">Azimuth</td></tr>
<tr><td>Mode A Code</td><td>1</td><td>1</td><td>X</td><td>SPI</td><td>Mode A Confidence</td><td>0</td><td>0</td><td>X cf</td><td>SPI cf</td></tr>
<tr><td>Altitude*</td><td colspan="4">Unused (4)</td><td colspan="3">Altitude Confidence</td><td colspan="2">Altitude Type (4)</td></tr>
<tr><td colspan="5">Special Bits**</td><td>(3) No. of Mode A Replies</td><td>(3) No. of Mode C Replies</td><td colspan="3">(Correlating Track) -0-</td></tr>
<tr><td>Mode 2 Code</td><td colspan="4">No. of Mode 2 Replies (4)</td><td colspan="3">Mode 2 Confidence</td><td colspan="2">Unused (4)</td></tr>
</table>

*Altitude is in Gray Code if any bit is low confidence-in flight level if all are high confidence.

**Special bits of interest:

Bit 3 - Boresight Indicator
Bit 6 - Swap Candidate

See Figure 3-2 for a list of other bits.

Figure 4-15: Target Report Format

At this point in time, the mode C code of a target report is still in its encoded Gray code format. If all the code bits are labelled as high confidence, this word can be converted to an integer flight level form, which is the form desired for display and other "human" uses. The conversion algorithm which has been designed for this purpose is developed in Appendix A. If some low confidence bits exist, however, the conversion is not attempted, as nonsense results could be obtained if any bit were set incorrectly.

The target declaration process for a reply entry is completed by performing t o bookkeeping actions. The first is the elimination of the sort table entry for the reply so that future replies will not attempt correlation with it. This is accomplished by nulling all the fields of the entry except the linkage pointer (see Figure 4-5), which is still required for bin searches. Future replies accessing the bin, upon finding this inactive entry, will remove it from the chain. The second action, required whenever a multiple hit report has been declared, is the nulling of the last two fields of the second reply data block. This action will insure that when that reply is later checked for target declaration, it will be passed over.

A flowchart of the target declaration process for a mode A or C sweep is presented in Figure 4-16.

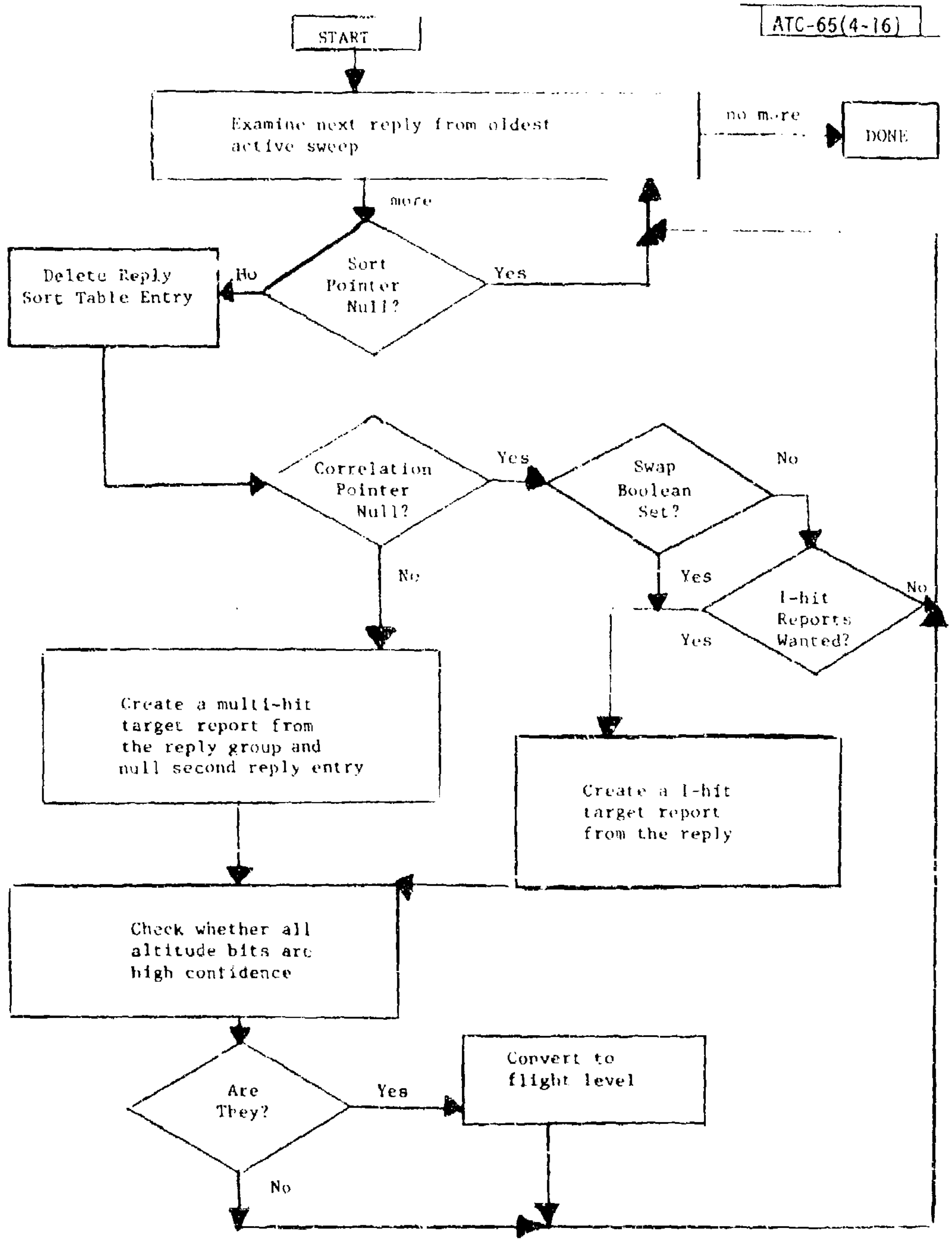


Figure 4-16: Target Declaration Flowchart

5.0 DISCRETE CODE CORRELATION

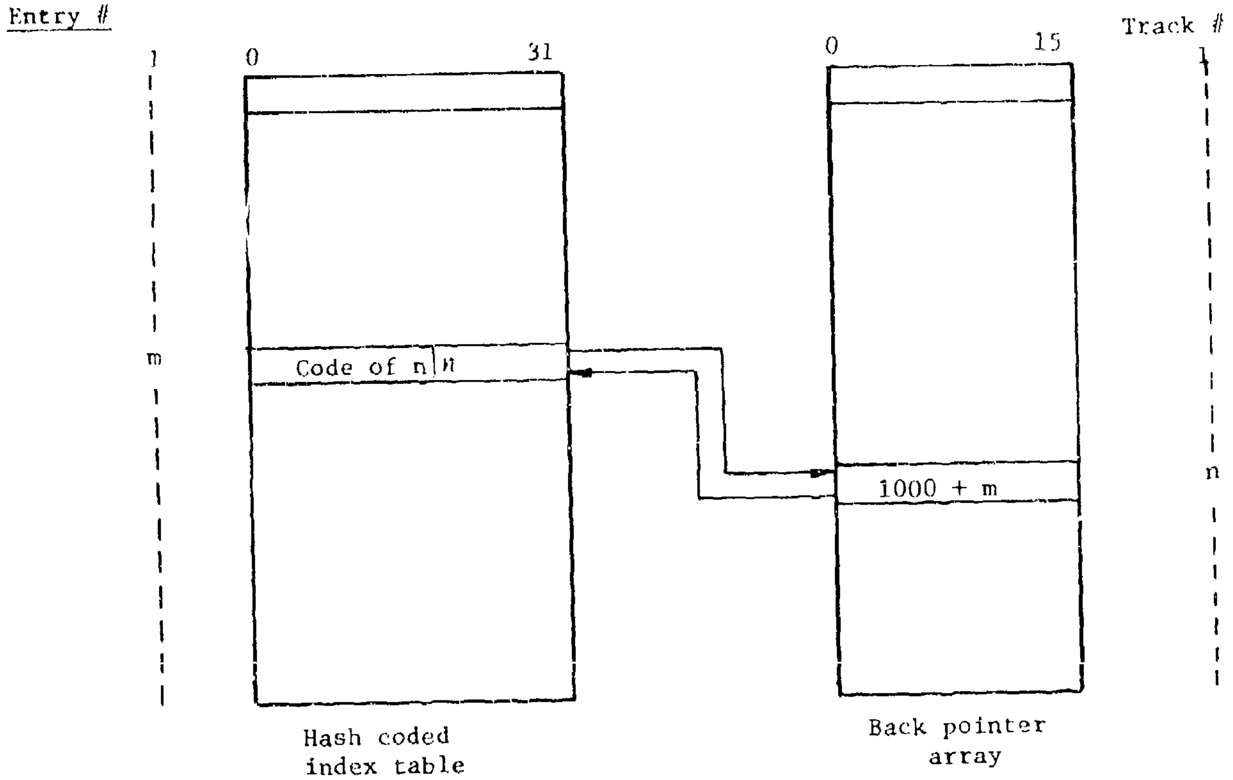
The ATCRBS system employs two types of mode A identity codes: discrete codes, which are uniquely assigned to aircraft, and non-discrete codes, which are used by all aircraft in the same flight situation (such as descending IFR). If we represent a 12-bit identity code by the four octal digits ABCD, as described in Chapter 2, then a code is non-discrete if C=0 and D=0 and is discrete if any of the C or D bits is a 1. Thus, non-discrete and discrete codes are often referred to as 64 and 4096 codes respectively to indicate the number of available codes of each type. Since discrete codes are assumed to be unique, the presence of the same one in both a target report and a track should be a sufficient criterion for correlation. To permit the rapid identification of the matching track, a hash coded index table is maintained that allows the accession of a track through its discrete code.

Unfortunately, the assumption that discrete codes are unique is not always true. Code assignment errors or reflection false tracks could result in the existence of more than one track for a given code; reflections, correlating fruit, or ringaround could lead to multiple target reports with the same code on the same scan. Thus, ambiguous correlation situations could arise that require the full-fledged processing used for non-discrete codes. Even in the normal case, dealing with only one target report and one track at a time, no assurance can be given that the report is valid. Thus, satisfaction of a set of position and altitude reasonableness tests is required before a discrete correlation is accepted. If these conditions are not met, the procedures for non-discrete tracks, described in the next two chapters, are again required.

5.1 Discrete Code Hash Table

All ATCRBS track information, for both discrete and non-discrete tracks, is physically located in the same track file. To permit the accession of discrete tracks through their identity codes, a separate hash coded index table is maintained in the system. In addition, a back pointer array is defined which both acts as an extension of this table and provides the information necessary for its dynamic manipulation. Figure 5-1 illustrates the use and interaction of these two entities.

The first track initiated for any discrete code has an entry created for it in the index table at the location determined by the hashing scheme described below. This entry references the track number, while the back pointer element for the track contains the value 1000 + the table entry number. Each time an additional track is created with this same discrete code, the hash table entry is changed to reference the new track number, and the back pointer element for the new track is made to point to the previously referenced track (refer to Figure 5-2). Thus, starting from either the hash table entry or any track in the loop, it is possible to determine all tracks possessing the same discrete code. The pointer to the table index can be distinguished from a pointer to another track since it will be the only one whose value exceeds 1000.



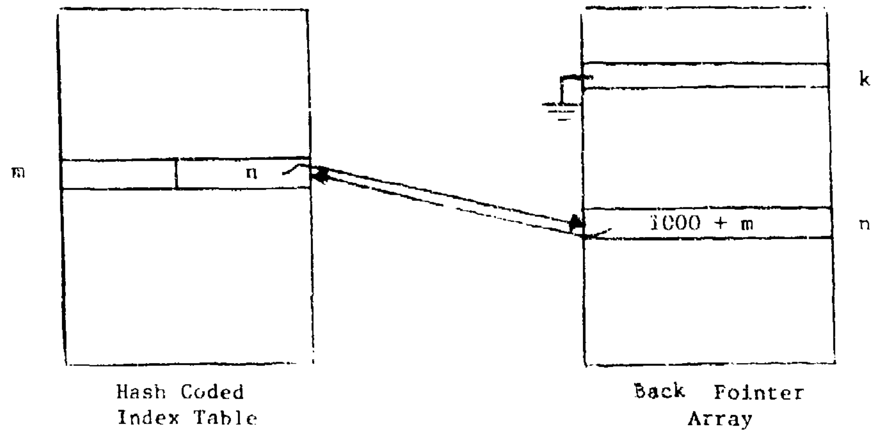
Hashing function, applied to discrete code sought, leads to index table entry m.

Entry m contains code represented (to indicate success of search on code) and pointer to track n.

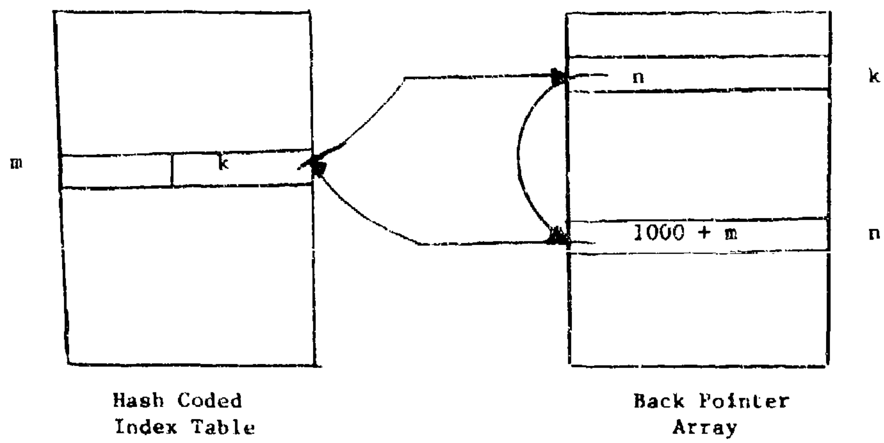
Back pointer element corresponding to track n contains 1000 + entry number m.

Figure 5-1: Discrete Code Data Structures

ATC-65(5-2)



Before inserting track k, with same code as track n (or after deleting track k).



After inserting track k (or before deleting it)

Figure 5-2: Dynamic Change of Hash Table System

Whenever a discrete track is dropped from the system or changes its code, it must be removed from this hash index system, although in the latter case it may be immediately re-entered in a different slot. The deletion algorithm, also shown in Figure 5-2, is simply the inverse of the insertion one. If the track's back pointer references a table element, and this element references the track, the track is the unique one with its code and so the table element is merely inactivated. However, if either of these premises is false, other tracks with the same discrete code remain in the system. By following the chain of pointers beginning with the track's back pointer, the loop consisting of the other tracks and the table entry can be traversed. When the entity preceeding the subject track is discovered, its pointer is set to the value of the dropped track's back pointer.

The hashing scheme chosen for the index table is open addressing. With this discipline, the index value of the element to be used for a track's entry is computed from its discrete code as described below. If that element is occupied, however, the first available higher numbered location is employed instead (with the first location considered to follow the last one). Conversely, a search for the existence of a track with a given discrete code begins at the computed hash address and proceeds linearly until either the desired track is located or an empty location is encountered; in the latter event, the search has failed and no such track exists.

The major complication in an open addressing scheme involves the deletion process. If a freed element and a never used element were indistinguishable, it is possible for a track entry to become detached from its hashed location, and thus be unlocatable during a search. Thus, as illustrated by Figure 5-3, two types of available elements are required: freed and never used. An insertion can be made in either type of element, but a search is over only when an element of the latter type is encountered. Since the existence of freed elements tends to lengthen searches, they should be coverted to the never used category whenever possible. The rule that applies is that any freed element that preceeds an unused one can be converted to unused. An occasional backwards stepping through the entire table, implementing this rule wherever possible, serves to produce the desired effect.

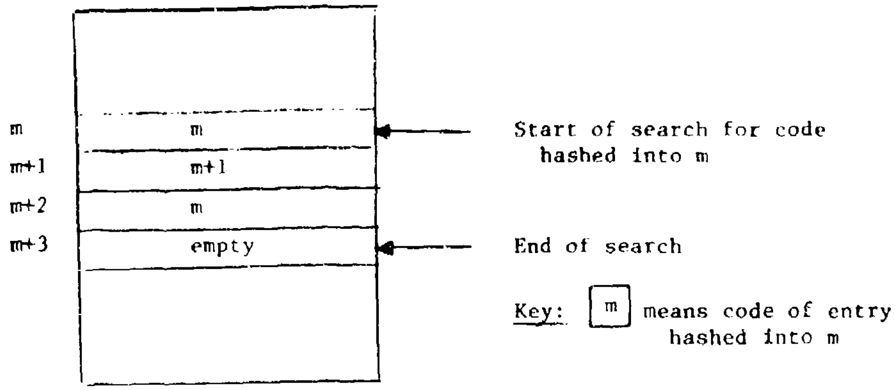
The other potential problem for open addressing is long searches whenever the table nears capacity or several tracks hash into the same area of the table. To prevent the first effect, the size of the table is set to twice the number of allowable ATCRBS tracks. That is, if N is the track upper bound:

$$\text{table size} = 2^m, \quad 2^{m-2} < N \leq 2^{m-1}$$

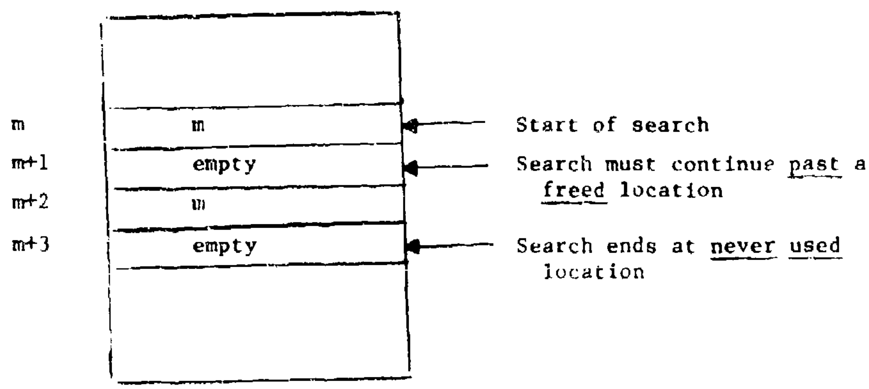
The table is sized as a power of two because the hash address rule selected uses the bits of the discrete code. The rule is:

$$\text{hash address} = 2 * [\underbrace{\ldots C_1 D_4 D_2 D_1}_{m-1 \text{ lowest code bits}}]$$

ATC-65(5-3)



Situation when two different codes hash into location m: second one is placed into first available slot above m, which was m+2 in this case.



Situation after entry m+1 is deleted; a new entry can be placed into either empty slot, but the search must only end at the never used one if second m entry is to be found.

<u>Figure 5-3:</u> Empty Hash Table Slots

The low order bits were chosen because discrete codes are often chosen in sequence; the factor of two greatly helps to spread out the table entries and prevent the bunching problem mentioned above.

5.2 Initial Target Processing

Before any target report enters into correlation, it must pass through three preliminary processing functions:

1. Target report reconstruction

2. Range sorting

3. Determination of correlation method to be employed (discrete or general)

The internal delay time for an aircraft transponder, between the receipt of the last pulse of an interrogation and the transmission of the first pulse of the reply, is specified to be identical for both modes A and C. Should the inter-mode delay variation exceed a critical value, the perceived range difference between mode A and mode C replies will prevent successful reply correlation. Thus two reports, one with only mode A replies and the other with only mode C replies, will be created for the aircraft. Since the symptom of such an out-of-spec transponder is so unmistakable, it is simple to correct the resulting error. It is clear that 1-hit reports, if permitted in the system, will always consist of only one mode. To prevent the formation of many correlating fruit reports, such reports are not permitted to enter into this reconstruction process.

The multiple hit target reports for the current sector are examined in order. If one is encountered whose number of replies for mode A field (mode C field) is zero, its number is placed on list 1 (list 2). At the end of this process, if both lists have one or more entries, each report on list 1 is compared in position with each report on list 2. Pairs are sought that satisfy:

$$|\Delta\rho| \leq 10 * \Delta\rho_{max}$$

$$|\Delta\theta| \leq \Delta\theta_{max}$$

where $\Delta\rho_{max}$ and $\Delta\theta_{max}$ are the reply correlation parameters. Whenever such a pair is located, a single report is created from the individual reports as follows:

$$\rho = \frac{\rho_1 + \rho_2}{2}$$

$$\theta = \frac{\theta_1 + \theta_2}{2}$$

mode A code and confidence:	use report 2
altitude, confidence, and type:	use report 1
special bits:	AND the two reports
# mode A hits:	use report 2
# mode C hits:	use report 1
mode 2 code and confidence:	combine reports 1 and 2 (update equations of 4.6)
# mode 2 hits	add reports 1 and 2

The two reports are then removed from the lists and the attempted pairing continues. After all pairs have been checked, the remaining reports in the target buffer are moved up to fill the holes left by the discarded reports.

After the reconstruction process is completed, all target reports for the sector, newly declared ones as well as those carried over from the previous sector, are entered into a range sort table. This table will be used by both the target to track correlation and radar reinforcement algorithms to permit rapid accession of targets in a given geometric area. The sort table, as shown in Figure 5-4, consists of a primary area and an overflow area. The entry for the first report in each range quantum is placed in the primary word assigned to the quantum. The entries for succeeding reports in a range quantum are placed in the overflow area, and are located by following the pointer chain emanating from the primary word. The sort bin to use is given by:

$$B = \frac{\rho}{Q} + 1 \quad \text{integer division}$$

where Q, the quantum size in miles, will be a function of the traffic load.

Each sort table entry contains the number of the target represented, a pointer to the next entry in the same range quantum (if any exists), and the target azimuth. The first two items are required, while the azimuth field permits a rapid check on whether the target is one of the ones sought. Thus, both a coarse range check (residence within the proper quantum) and a fine azimuth test can be performed on a report without the need to access its data block.

Once all reports are sorted, each in turn is checked to determine whether it can undergo discrete correlation. The following conditions must all be met for this process to be employed:

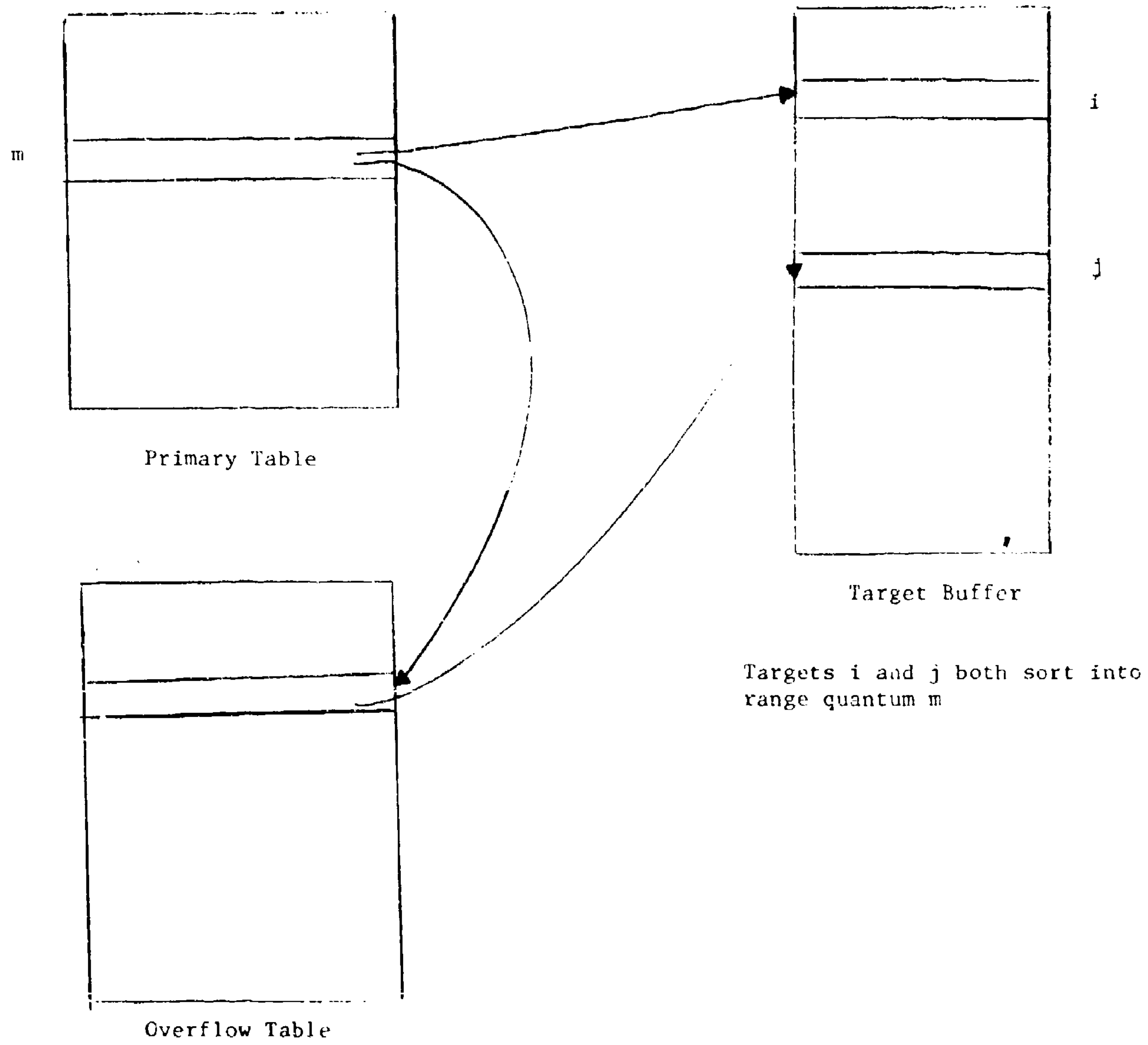
1. The target has a discrete 4096 code.

2. All code bits of the target have been declared with high confidence.

3. At least one track exists with the same discrete code.

4. At most one real track exists with the same discrete code.

The first condition is obvious. The second eliminates from consideration reports whose code is not known with certainty. As low confidence bits are often wrong, the proper code on which to correlate cannot be determined.



0–7	8–15	16–31
Target Number	Next Bin Entry	Target Azimuth

Sort Table Entry

ATC-65(5-4)

Figure 5-4: Target Sort Table

The third and fourth conditions are checked through reference to the discrete code hash table described above. A search is initiated on the report's discrete code, and the number of tracks found is noted. If none, discrete correlation is impossible; if one, everything is proper; if two or more, something suspicious has occurred. The only system feature that should lead to two tracks with the same discrete code is reflection false targets (refer to Chapter 10). Thus, if all but one of the tracks are labelled false, discrete correlation is permitted to continue. However, if two or more are real, the more powerful non-discrete correlation algorithms are used, as they are more capable of dealing with unusual system behavior.

5.3 Discrete Correlation Procedure

In the normal case, discrete correlation will deal with only one target report and track at a time. If their positions are reasonably near each other and they agree on altitude, the report and track will be correlated. However, as noted in the chapter introduction, numerous special cases must be identified and treated within the overall discrete correlation process.

The main components of this correlation algorithm are the following:

1. For each discrete coded report, determine how many tracks with matching code agree in position and altitude

   0 - revert to general correlation
   1 - proceed
   $\geq$2- revert to general correlation

2. Determine whether ringaround may be present; if possible, revert to general correlation

3. If 2 or more reports associate with the same track, choose the proper report for correlation

4. Determine whether to correlate in the current sector or to delay correlation to the subsequent sector

The remainder of this section will elaborate on these ideas. A flowchart of the actions to be presented is provided by Figure 5-5.

As discussed in the previous section, a report is occasionally allowed to enter discrete correlation even if more than one track matches its code. Thus, the proper track to choose must be determined. In addition, as shown in the introduction, even if only one track exists, the correlation may be improper, as the report itself could be invalid. Thus, all matching tracks must be checked to determine whether any is reasonably close in position and altitude to the report to create an acceptable correlation. The tests performed for each track relative to the report are:

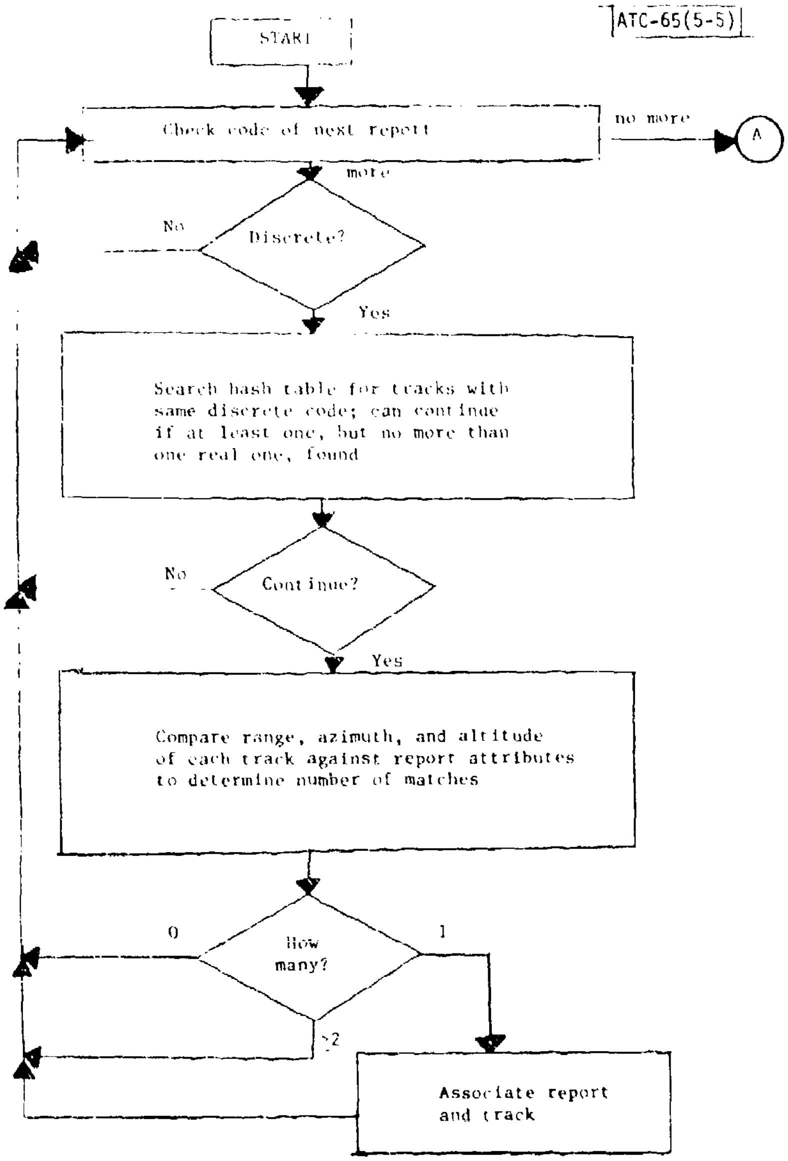


Figure 5-5: Discrete Code Correlation Process (1 of 2).

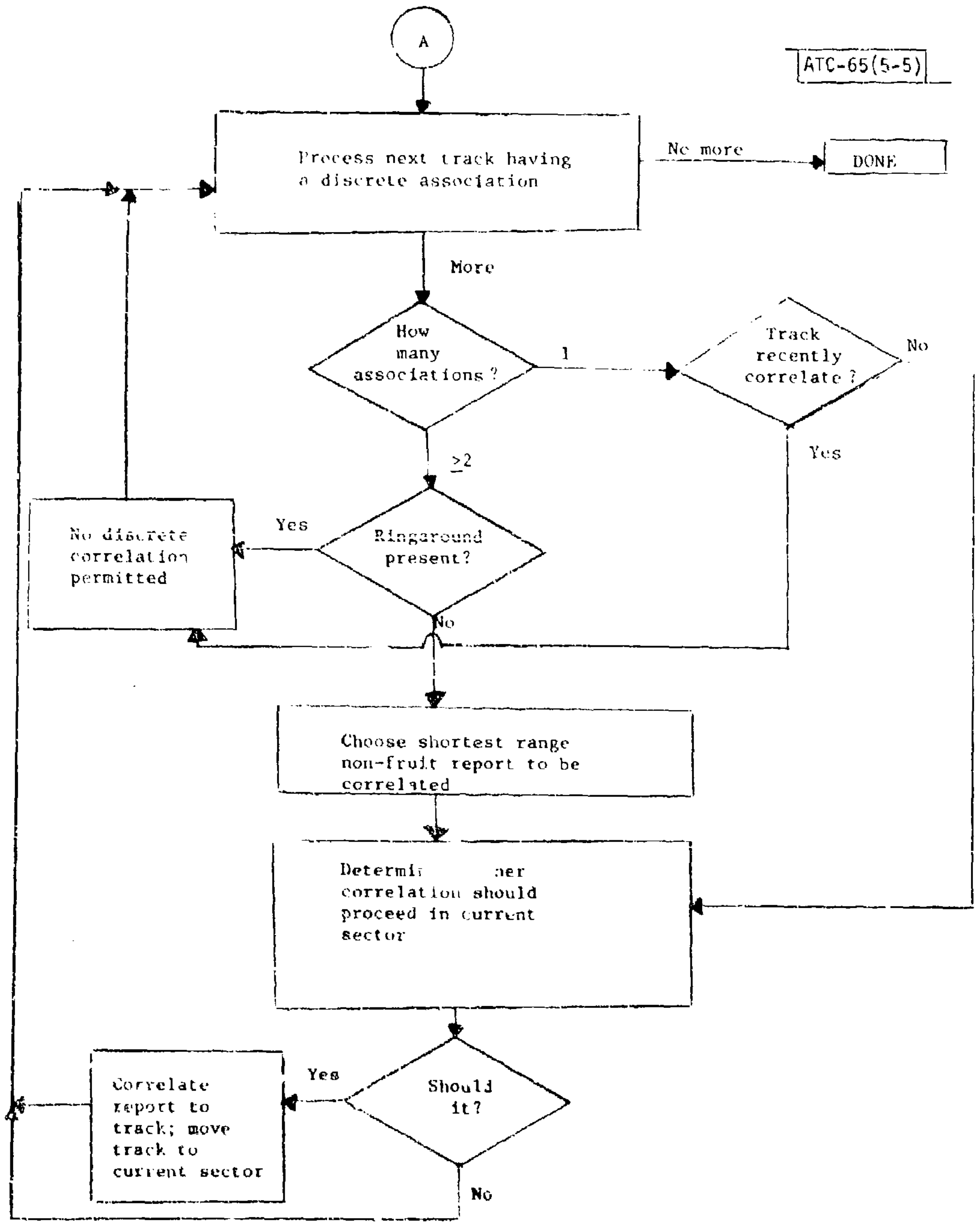


Figure 5-5: Discrete Code Correlation Process (2 of 2).

1. azimuth

   (a) for reports with range $> \rho_{disc}$

   $$|\Delta\Theta| \leq \Delta\Theta_{disc}$$

   (b) for reports with range $\leq \rho_{disc}$

   no test required: $|\Delta\Theta| < 30^\circ$ is acceptable this close to the sensor and $|\Delta\Theta| \geq 30^\circ$ is covered by 2(b)

2. range

   (a) for target/track pairs with $\Delta\Theta < 30^\circ$

   $$|\Delta\rho| \leq \Delta\rho_{disc}$$

   (b) for target/track pairs with $\Delta\Theta \geq 30^\circ$

   $$\rho^2_{targ,gnd} + \rho_{trk,gnd} - 2\rho_{targ,gnd}\,\rho_{trk,gnd} * \cos(\Delta\theta) \leq \Delta\rho^2_{disc}$$

   where gnd means ground range

3. altitude

   (a) if report is not a code swap candidate (see Chapter 4)

   $$\Delta h \leq \Delta h_{max}$$

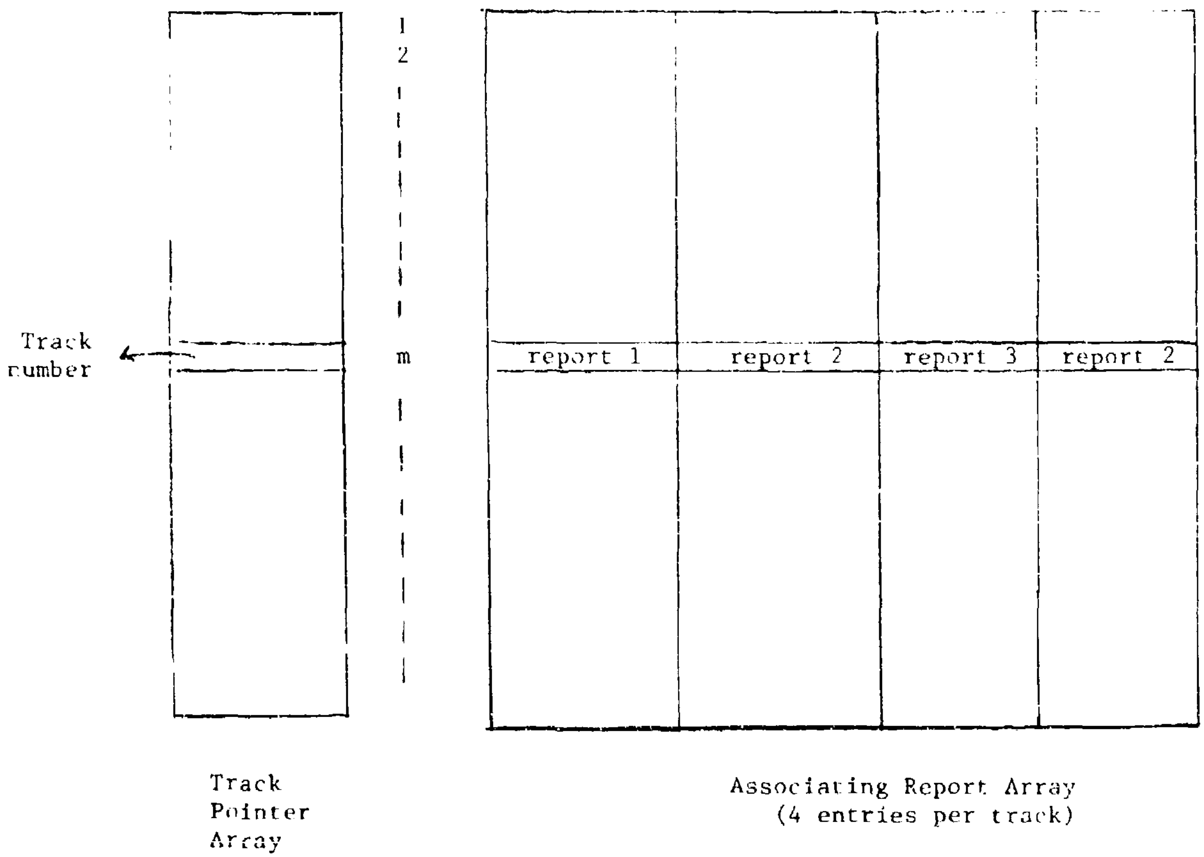
   (b) if report is a code swap candidate

   $$\Delta h \leq 1/2\Delta h_{max}$$

The calculation of Δh between a report and a track is described in the Appendix. The tighter bound on altitude for potential code swap reports permits the report to enter general correlation and undergo the code swap when a suspect altitude match exists.

If all three reasonableness checks are satisfied with one, and only one, track, an association is made as shown in Figure 5-6. Should no track be successful, the report enters general correlation to seek its proper track; should two or more tracks pass the tests, general correlation is needed to apply a more complex set of criteria to the situation. After all reports have been processed, a full association table as depicted in Figure 5-6 will exist. Tracks with only one associating report are matched with that report, but tracks with two or more associations must still undergo a selection process.

ATC-65(5-6)



Each new track receiving a discrete code association is assigned the next available row in the structure.

Each additional association is placed in the next available column (up to 4).

Figure 5-6: Discrete Code Association Table

Ringaround occasionally occurs when the elevation angle of an aircraft exceeds 30^o. Its symptom is several targets with the same code at the same range but at different azimuths. Thus, if more than one target in a sector associates with the same discrete track, and the track has an elevation angle above 30^o, the determination of the "real" target must be left to the more complex general correlation procedure. In addition, if any track with such an elevation angle has only one association, but the track has just recently correlated (within half a scan), a wider azimuth ringaround is probably occurring. Thus, again correlation is not performed by the discrete algorithm.

Any other situation in which two or more discrete reports associate with the same track is probably caused either by ground reflection producing shadow reports, or by the coincident correlation of a fruit reply from the same aircraft. In the former case, the shortest range report will be the real one, while in the latter case the fruit report will almost always consist of one reply of each mode (see Chapter 10). Thus, the correlation rule chosen in multiple discrete association cases is that the shortest range, non-2-hit A/C report, is to be correlated with the track. The remaining reports are then labelled false and not allowed to enter into general correlation.

The final issue in discrete correlation, after a target/track pair has been selected, is whether to perform the correlation in the current sector or postpone this action to a future sector. The latter choice is preferable whenever the track has a reasonable expectation of locating a more valid report in a subsequent sector; this hope would occur when the predicted sector for the track is subsequent to the current sector. The rules which implement this idea are the following:

1. If the track's predicted sector is the current sector or a previous sector, correlate immediately.

2. If the track's predicted sector is a subsequent sector, but the target report has been held as long as possible in the system and must be output this sector, correlate immediately rather than lose the chance (another report with the same discrete code is always doubtful).

3. If the track's predicted sector is a subsequent sector, and the target report can be delayed another sector, postpone the correlation decision and hold the report for the next sector (in the manner described in Chapter 7).

If the correlation is accepted in the current sector, the track number is placed in the proper field of the target report and the target number entered into the track file. The only other action required arises if the track is not resident on the linked list for the current sector (refer to Chapter 9). Track update cannot process any such track; thus the track must be removed from its present list and placed at the end of the list for the current sector.

6.0 TARGET TO TRACK ASSOCIATION

All target reports in the current sector that were not discretely correlated, either because their identity codes were not discrete or because they failed to meet one of the criteria of the discrete algorithm, undergo a more complex general correlation procedure. This process has two components: association, which identifies all possible pairings of targets and tracks, and correlation, which chooses from among these the proper track for each report. This chapter will discuss the former of these actions.

In order for a target report and track to successfully associate, they must lie reasonably close to each other in three dimensional space. That is, their differences in range, azimuth, and altitude must be smaller than the largest expected track prediction error. In addition, agreement in identity code is desirable, although the possibility of code reassignment during flight precludes this being a strict requirement.

During the association process, reply correlation errors may come to light. When such an erroneous target report is identified, "code swapping" is employed to reconstruct the proper pairings of mode A and mode C codes. This process requires the presence of certain 1-hit target reports, namely those carefully preserved during the target declaration process of Section 4.6.

The next four sections of this chapter discuss all of the key concepts of the association process. The last section then ties all of these ideas together and presents the overall association algorithm.

6.1 Association Cross Reference Table

The most important association data structure is the track/target cross reference table. This table has an entry for each association pair identified during the association process that specifies the track number, target report number, and score of the pairing. In addition, the table permits the easy identification of all reports associating with a given track and of all tracks associating with a given target.

Conceptually, this table can be represented as shown in Figure 6-1. Each entry contains four fields: track number, target number, score, and next entry pointer. All pairings for any given track are located contiguously, while all pairings for a target are linked together through the pointer field. In addition, each track and each target has a separate pointer to its first entry.

The actual storage implementation chosen for these table entries is presented in Figure 6-2. Three two-dimensional arrays are employed, which contain, for any given index (i, j), the target number, score, and next entry

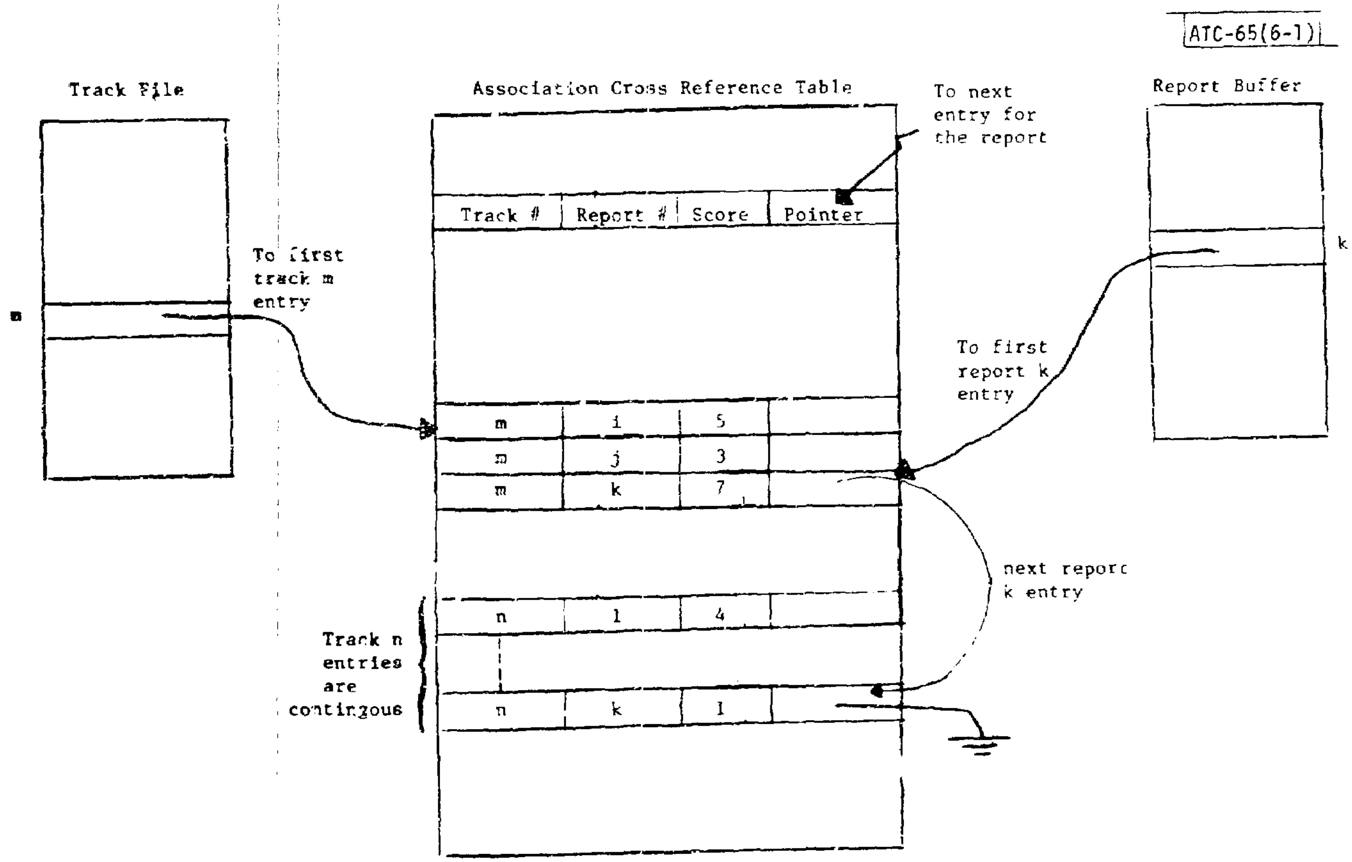
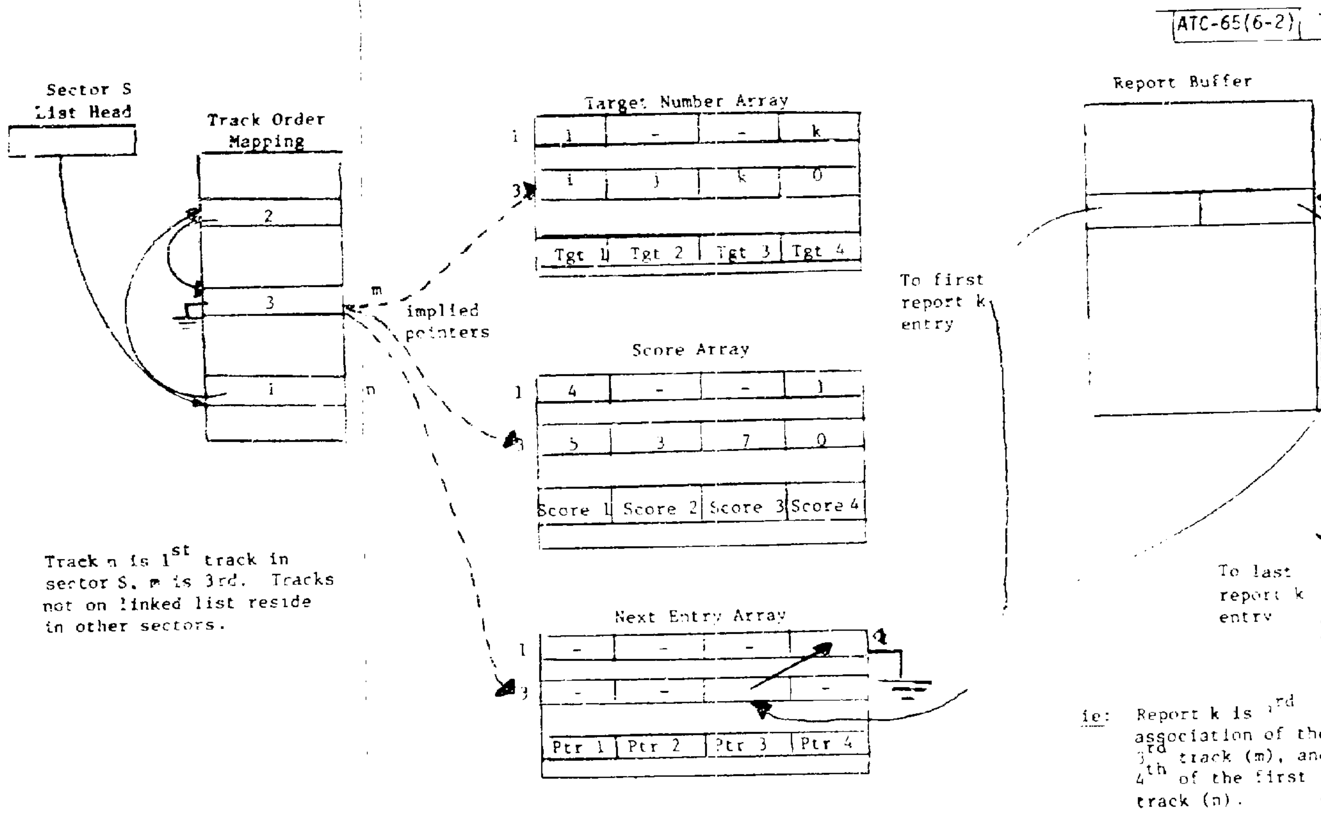


Figure 6-1: Conceptual View of Association Table.



Same association example as in Figure 6-1

Figure 6-2: Actual Association Table Structure.

pointer respectively of the entry corresponding to that index. The first subscript i of each array ranges from one to the maximum number of tracks that can exist in a sector, and indicates the entry is for the i^{th} track in the sector (the tracks, as shown in the figure, are ordered by a linked list structure). The second subscript j ranges from one to the maximum number of associations permitted for a track, and indicates that the entry is the j^{th} for the track.

The mapping from track numbers to sector order numbers for the first subscript permits a significant reduction in the size of these arrays, as only a small fraction of all tracks can reside in one sector. The restriction of a limited number of associations for any track, which is a feature of this implementation but not of some alternative ones, was felt to be desirable as it provides the system designer with some control over the performance of the overall correlation algorithm. For example, by reducing this limit, many fewer interlocking association situations will arise that correlation must resolve. This may decrease execution time noticeably with slight system performance degradation. Thus, an optimum limit can be sought. Also, by placing a limit on the number of associations allowed for a track, a track is permitted to be coasted when all of its best reports are correlated with other tracks, even when other lower quality reports exist. This could well prevent some serious correlation errors.

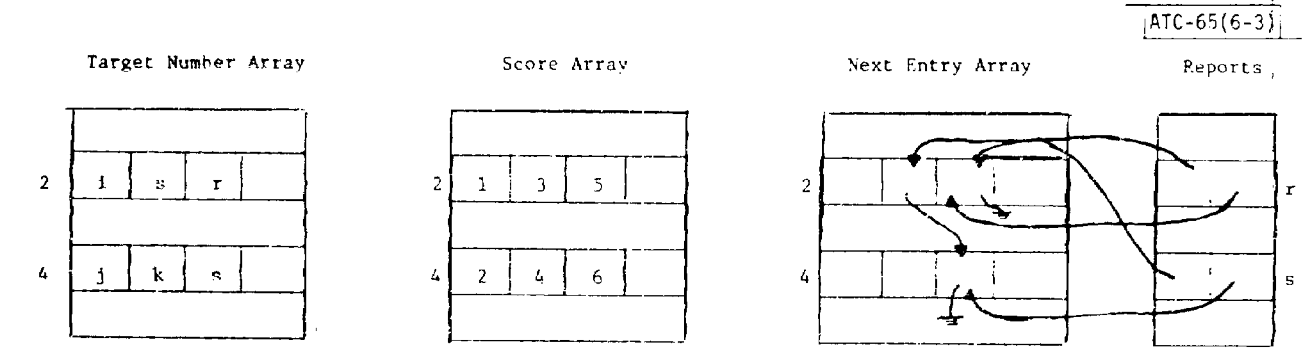
The target number and score arrays for an index directly contain these items for the corresponding entry. The next entry pointer array, however, requires some decoding of the value stored. In particular, if the next entry for the target report is the r^{th} entry for the k^{th} track, then:

stored value = $M \times k + r$

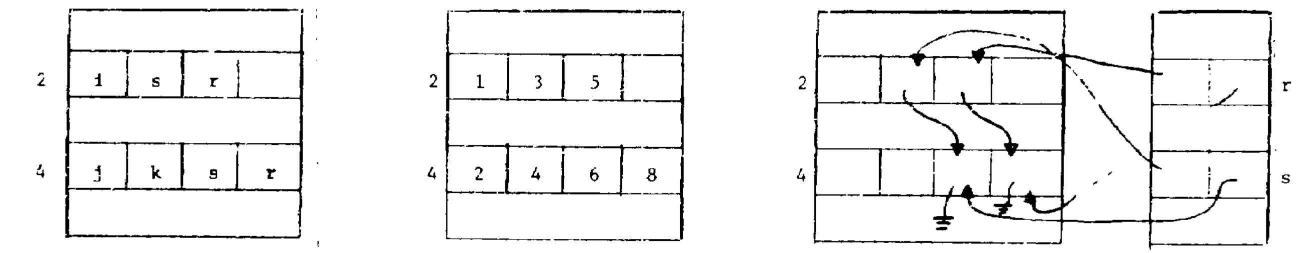
M = maximum number of associations per track

Thus, integer division by M of one less than this value provides the first subscript for the next entry, while a simple subtraction provides the second subscript. In addition, each report has similarly encoded pointers to its first and last entries.

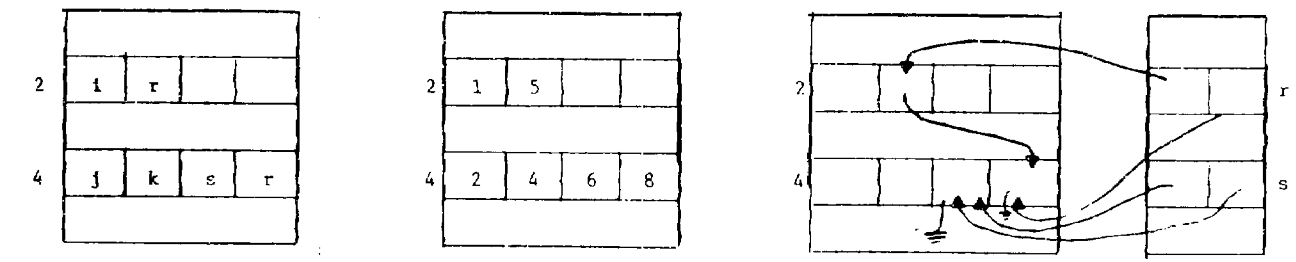
To create an association entry for track k and report j, the sequence number i of the track is first determined from its position in the sector linked list. The entry itself is then placed into the (i, j) elements of the three arrays that constitute the table. The value of the next entry array element is set to zero, as the new entry is always made the last one for the report. To accomplish this, the previously last entry for the report, specified by the report's last entry pointer, is set to point to the new entry, and then the last entry pointer itself is set to this same value. Figure 6-3 illustrates this sequence of events.



(a) initial state



(b) report r was added to the 4th track (r's pointer chain is altered)



(c) report s was deleted from the 2nd track (s's and r's pointer chains are altered)

Figure 6-3: Dynamic Change of Association Table

When an association entry must be discarded, for one of the reasons specified in section 6.5, the three actions depicted in Figure 6-3 must occur. Assume the entry to be deleted is for target j and the i^{th} track in the sector. The first action is to link the report j pointers around this entry. Starting with the report's first entry pointer, the pointer chain is traversed until the entry in row i of the table is encountered. The pointer of the previous entry is then set equal to that entry's pointer. Also, if the deleted entry was the last for the target, its last entry pointer is adjusted. The second action is to move the last association for row i into the vacated slot, as holes would cause problems later. Finally, the pointer chain for the target contained in this moved entry is updated to reflect its new position. As before, this is done by finding the prior entry and altering its pointer field.

To find all reports associated to track i, all entries in the i^{th} row of the target array are examined. To identify all tracks associating with a report, the report's pointer chain is traversed and decoded.

6.2 Association Parameters and Types Matrix

Each potentially associating target and track pair, identified as described in Section 6.5, is examined to determine the level of agreement on the three key attributes: geometric position (range and azimuth), identity code, and altitude. Depending upon the results of these tests, the pair will form one of the following types of association:

1. Sure association - the pair is accepted

2. Potential association - further tests are required on the pair

3. Potential code swap - a possible reply correlation error has been found

4. No association - the pair is rejected

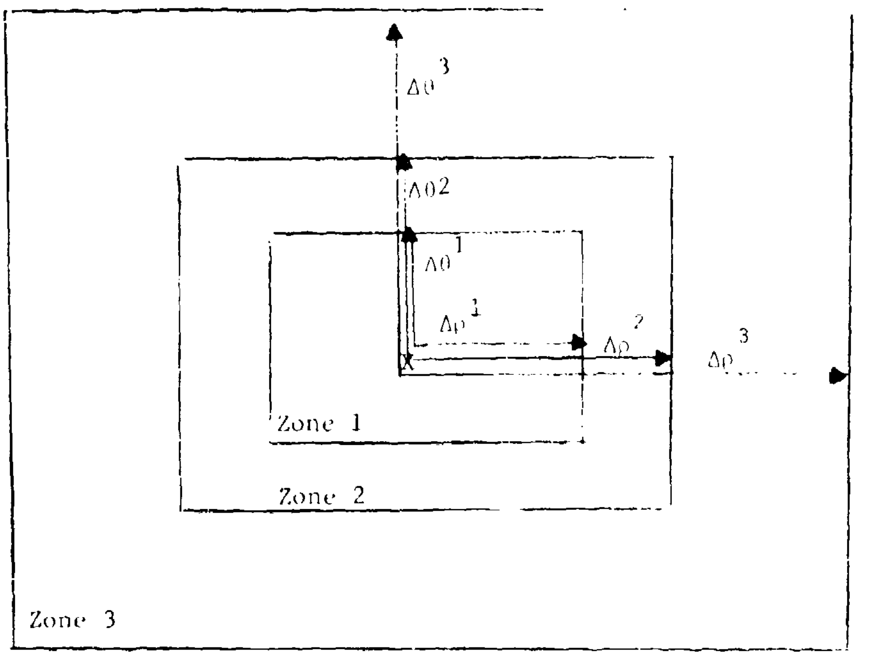
The entire issue of code swapping is examined in Section 6.4.

The first test made on the target/track pair is range and azimuth agreement. Three boxes are constructed around the predicted track position, as shown in Figure 6-4. The sizes of these boxes meet the following conditions:

1. If the tracked aircraft is flying in a straight line, the target report will fall in the smallest box, thereby creating a zone 1 association.

2. If the tracked aircraft is turning normally, the target report will fall at worst in the middle sized box, thereby creating a zone 2 association.

ATC-65(6-4)



X is predicted track position

Figure 6-4: Association Zones

3. If the tracked aircraft is maneuvering abnormally, or if the track has been detoured by an erroneous correlation, the target report will fall at worst in the largest box, thereby creating a zone 3 association.

If the report falls outside the largest box, the association is rejected. Otherwise, the association is labelled with the proper zone value and testing continues.

The method for deriving the formulas for the zone 1 box, presented in Figure 6-5, is to determine the largest possible straight flight error in range or azimuth by assuming the worst case errors for the previous two data points (since tracking is done by two point interpolation, earlier points are irrelevant). The track firmness f and history firmness g, which give the number of scans since the last correlation and between the last two correlations respectively, are maintained in the track file (refer to Figure 8-6). Then, if the assumption is made that at close range the azimuth accuracy in feet cannot exceed the range accuracy (to prevent the box from shrinking to zero), the resulting formulas are:

$$\Delta\rho^1 = d_\rho \times \left[1 + \frac{2f}{g}\right] \qquad \text{(n. miles)}$$

$$\Delta\theta^1 = \text{Max}\left\{d_\theta, \frac{d_\rho}{\rho}\right\} \times \left[1 + \frac{2f}{g}\right] \qquad \text{(radians)}$$

where d_ρ = report range accuracy (n. miles)

d_θ = report azimuth accuracy (radians)

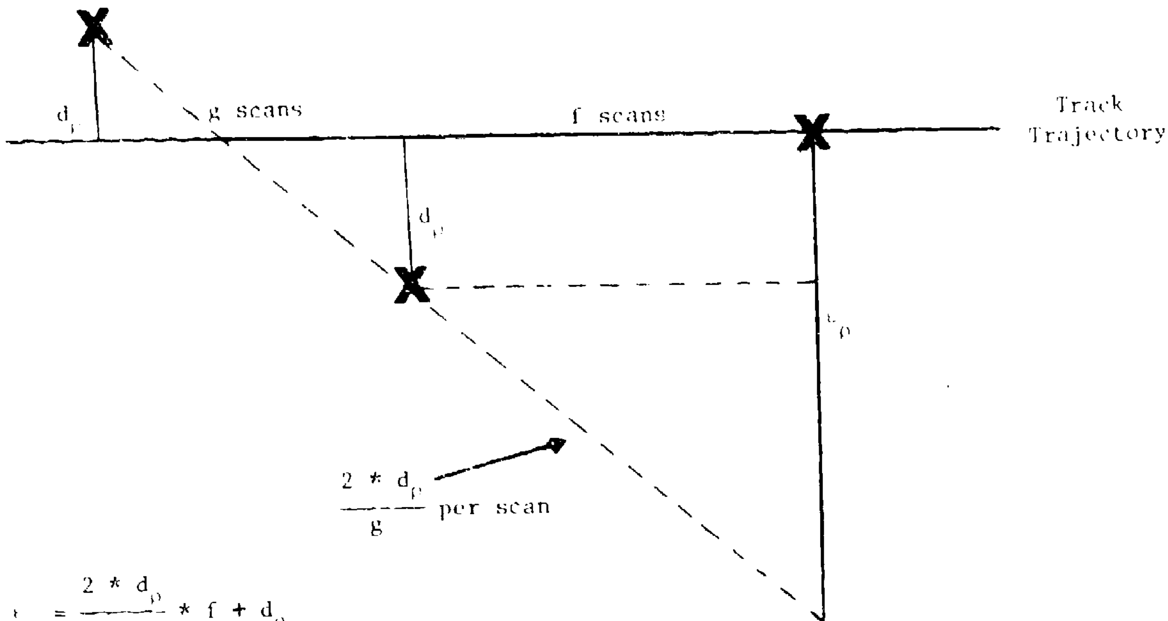
ρ = predicted track range (n. miles)

The zone 2 box dimensions are calculated by assuming the aircraft being tracked is in a circular turn. Figure 6-6 depicts the assumed configuration, and presents the derivation of the required formulas for the worst error case. As seen, each formula has two terms: one depending upon report accuracy which is identical to the box 1 relation, and the other depending upon the turning acceleration rate. Since the latter error component is always in miles, rather than in degrees, the resulting formulas become:

$$\Delta\rho^2 = \Delta\rho^1 + 0.05 \times \left[f^2 + fg\right] \times a_g \qquad \text{(n. miles)}$$

$$\Delta\theta^2 = \Delta\theta^1 + \frac{0.05}{\rho} \times \left[f^2 + fg\right] \times a_g \qquad \text{(radians)}$$

where a_g = turning acceleration (g units) and a second scan is assumed.



$$\varepsilon_\rho = \frac{2 * d_\rho}{g} * f + d_\rho$$

$$= d_\rho \left[1 + 2\frac{f}{g}\right]$$

similarly,

$$\varepsilon_\theta = d_\theta \left[1 + 2\frac{f}{g}\right]$$

assume θ error cannot become less in feet than ρ error

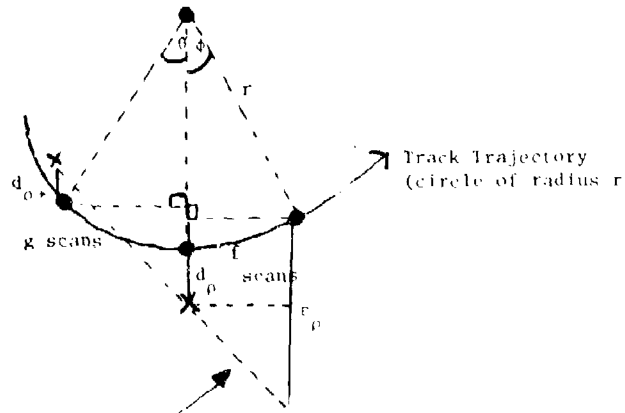
$$\varepsilon_\theta = \text{Max}\left\{d_\theta, \frac{d_\rho}{\rho}\right\} * \left[1 + 2\frac{f}{g}\right]$$

$\uparrow$
ρ error

ATC-65(6-5)

Figure 6-5: Zone 1 Size Determination

ATC-65(6-6)



$$\frac{d_\rho + (r - r\cos\theta) + d_\rho}{g} \text{ per scan}$$

$$\epsilon_\rho = \frac{2d_\rho + r - r\cos\theta}{g} * f + (r - r\cos\phi) + d_\rho$$

$$1 - \cos x \approx \frac{x^2}{2}$$

$$\therefore \epsilon_\rho \approx \frac{2fd_\rho}{g} + \frac{rf\theta^2}{2g} + \frac{r\phi^2}{2} + d_\rho$$

$$r \approx \frac{V^2}{a} \text{ and } r * \text{angle} = V * \text{scans} \quad \begin{cases} a = \text{acceleration} \\ V = \text{velocity} \end{cases}$$

$$r * \text{angle}^2 = a * \text{scans}^2$$

$$\therefore \epsilon_\rho = d_\rho \left[1 + \frac{2f}{g}\right] + \frac{a}{2}\left[fg + f^2\right]$$

$$a \approx .1 * a_g \text{ nmi/scan}^2 \quad \begin{cases} a_g = \text{g's of turn} \\ 4 \text{ second scan} \end{cases}$$

$$\therefore \epsilon_\rho = d_\rho \left[1 + \frac{2f}{g}\right] + .05\left[f^2 + fg\right] a_g$$

second term is in miles, even for 0 error

$$\therefore \epsilon_\theta = d_\theta \left[1 + \frac{2f}{g}\right] + \frac{.05}{\rho}\left[f^2 + fg\right] a_g$$

Figure 6-6: Zone 2 Size Determination

Since zone 3 is intended to account for unexpected maneuvers or for tracking errors due to previous incorrect correlations, no formulas can be derived to represent its size. Instead, it is simply defined as follows:

$$\Delta\rho^3 = n \times \Delta\rho^2$$

$$\Delta\theta^3 = n \times \Delta\theta^2$$

The value to use for n, and the largest size to permit this box to become, can only be determined empirically.

Except when the track is very near the sensor, the association zone for a target/track pair is found by comparing their range and azimuth differences against the box size variables $\Delta\rho^i$ and $\Delta\theta^i$. If the values $\Delta\rho^0$ and $\Delta\theta^0$ are set to 0, and $\Delta\rho^4$ and $\Delta\theta^4$ are defined to be infinite, the component zones are determined as follows:

$$z_\rho = i \text{ if } \Delta\rho^{i-1} < \Delta\rho \leq \Delta\rho^i$$

$$z_\theta = j \text{ if } \Delta\theta^{j-1} < \Delta\theta \leq \Delta\theta^j$$

The association zone then becomes:

$$z = \text{Max } \{z_\rho, z_\theta\}$$

The one exception to this rule occurs when the target is a general (not potential code swap) 1-hit report, which is possible for ATCRBS systems in very low fruit environments that permit such entities. In order to penalize the highly suspect report, the zone of its association is set to one higher than the calculated value.

If a track is very near the sensor, the values of $\Delta\theta^i$ exceed 360°, and thus the zone 1 azimuth comparison would always be satisfied. This would lead to the declaration of zone 1 for a target/track association in which $\Delta\rho$ was very small, even if the two entities were on opposite sides of the sensor and hence very far apart. To correct this problem, the zone test for tracks within a parametric range is replaced by:

$z = i$ if

$$\left[\Delta\rho^{i-1}\right]^2 < \left[\rho^2_{gnd,trk} + \rho^2_{gnd,tgt} - 2\rho_{gnd,trk}\,\rho_{gnd,tgt}\cos\Delta\theta\right] \leq \left[\Delta\rho^i\right]^2$$

where gnd means ground range. Again, this value is incremented for general 1-hit reports.

If the zone value calculated for an association is 1, 2 or 3, the testing can continue. However, any association whose zone is 4 or greater is immediately rejected.

The next association parameter checked for the target/track pair is mode A identity code agreement, for which the symbol ΔC will be used. First, the number of high confidence bit disagreements between the target and track codes is computed. Such a disagreement occurs whenever the two codes both have a high confidence declaration for a given bit position, but the values are opposite (a "0" versus a "1"). The number of such instances is given by the weight of the following syndrome sequence:

$$S = \overline{(A_k \oplus A_g)} \cup AC_k \cup AC_g$$

where A is mode A code sequence
AC is mode A confidence sequence
k refers to track
g refers to target

Total code agreement, denoted by $\Delta C = 0$, occurs when $||S|| = 0$, that is when all bits of S are zeroes. Should this situation occur, however, because the target report had no high confidence code bits, it will be called default code agreement instead, and the value of ΔC will be set to $1/2\Delta C_{max}$. The value of ΔC_{max} is irrelevant; the symbol is used only for parallelism with the altitude situation discussed below. The next possible case, potential code agreement, exists when fewer than a parametric number of bit disagreements are found. This number, typically set at one, is related to the reply processor error rate. Potential agreement, represented by $\Delta C = \Delta C_{max}$, thus occurs when $||S|| \leq N_{err}$. Finally, code disagreement between the association pair exists whenever more than the allowable number of bit disagreements are found. That is, this case, represented by $\Delta C = 2\Delta C_{max}$, occurs when $||S|| > N_{err}$. Examples of all of these situations are presented in Figure 6-7.

The final association condition between a target report and a track file that must be checked is the relative level of agreement of their respective mode C altitudes. If both altitude estimates were in flight level, this check would be trivial. In such a case, the level of agreement, denoted by the symbol Δh, would be computed as follows:

$$\Delta h = |h_k - h_g|$$

where h is altitude in flight levels (hundreds of feet).

However, either or both altitudes could be non-existent, brackets only (indicating no altimeter), or still in Gray code due to the presence of one or more low confidence bits. Thus, there are a large number of possible comparison situations. Appendix A details how the value of Δh is determined in all of these cases. The nomenclature for the type of agreement that exists for the pair is as follows:

Track Code and Confidence:

A_k: 010 001 100 011

AC_k: 000 000 000 000 (All High Confidence)

Target Code and Confidence:

Case 1:

A_g: 010 000 100 001

AC_g: $000\ \ 01\overset{*}{1}\ \ 010\ \ 0\overset{*}{1}0$

* = Disagreement with Track

$\Delta C = 0$, As All Differences Are Low Confidence

Case 2:

A_g: 000 000 000 000

AC_g: $1\overset{*}{1}1\ \ 11\overset{*}{1}\ \ \overset{*}{1}11\ \ 1\overset{*}{1}\overset{*}{1}$

$\Delta C = 1/2\ \Delta C_{max}$, No High Confidence Bits Disagree Because All Bits Are Low Confidence

Case 3:

A_g: 010 001 000 001

AC_g: $001\ \ 000\ \ \underset{\uparrow}{\overset{*}{0}}00\ \ 0\overset{*}{1}1$

$\Delta C = \Delta C_{max}$, One High Confidence Disagreement

Case 4:

A_g: 000 111 100 011

AC_g: $0\underset{\uparrow}{\overset{*}{0}}0\ \ \underset{\uparrow}{\overset{*}{0}}\overset{*}{1}0\ \ 010\ \ 000$

$\Delta C = 2\Delta C_{max}$, More Than One High Confidence Disagreement

Figure 6-7: Code Matching Examples

$0 \leq \Delta h \leq \frac{1}{2} \Delta h_{max}$: altitude agreement

$\frac{1}{2} \Delta h_{max} < \Delta h \leq \Delta h_{max}$: potential altitude agreement

$\Delta h > \Delta h_{max}$: altitude disagreement

The value of the parameter Δh_{max} has typically been set at 10, which represents a difference of 1000 feet between report and track.

As stated in Chapter 1, the track file does not maintain a mode 2 code. Thus, no mode 2 agreement calculation is possible, and mode 2 plays no role in association or correlation.

Once the geometric zone and values of ΔC and Δh have been determined for an association pair, the type of association that exists can be identified. Figure 6-8 presents the two matrices that supply this type information. The first matrix, shown in Figure 6-8(a), applies to all associations in which the target report is not a swap candidate. This status has been determined in reply correlation (see Section 4.6) and is indicated by the corresponding bit of the target report (refer to Figure 4-14). The second matrix, in Figure 6-8(b), is used by associations in which the target report is a swap candidate. The entries in which a dash appears are those for which the swap status is irrelevant; the corresponding entry in Figure 6-8(a) is applicable in both cases. Also, if the potential code swap in fact does not occur, the association type reverts to that indicated in Figure 6-8(a). The use of these matrices is discussed more fully in sections 6.4 and 6.5.

Six categories of association are defined in these matrices. The meaning of each type is as follows:

1. Perfect association - all attributes (position, code, and altitude) match fully

2. Acceptable association - the code or altitude attribute (or both) is suspect, but no further testing is deemed necessary due to the excellent positional agreement

3. Potential association - the combination of suspect code or altitude (or both) with suspect position requires the performance of the Velocity Reasonableness Test given in the next section

4. Potential code swap (alt $\overline{\text{code}}$) association - the report altitude, but not code, matches that of the track; since the report is paired with another, code swapping could improve this condition

5. Potential code swap ($\overline{\text{alt}}$ code) association - dual of 4

6. No association - attribute differences warrant rejection

	$\Delta h \le \frac{1}{2} \Delta h_{max}$	$\frac{1}{2} \Delta h_{max} < \Delta h \le \Delta h_{max}$	$\Delta h > \Delta h_{max}$
$\Delta C \le \frac{1}{2} \Delta C_{max}$	Perfect Perfect Potential	Acceptable Potential No	No No No
$\frac{1}{2} \Delta C_{max} < \Delta C \le \Delta C_{max}$	Acceptable Potential No	Acceptable Potential No	No No No
$\Delta C > \Delta C_{max}$	Acceptable Potential No	Acceptable Potential No	No No No

<u>Key:</u>

Zone 1 result

Zone 2 result

Zone 3 result

ATC-65 (6-8a)

<u>Figure 6-8(a):</u> Association Matrix for Non-code Swap Reports.

	$\Delta h \leq \frac{1}{2} \Delta h_{max}$	$\frac{1}{2} \Delta h_{max} < \Delta h \leq \Delta h_{max}$	$\Delta h > h_{max}$
$\Delta C \leq \frac{1}{2} \Delta C_{max}$	- - -	$(\overline{alt}\ code)$ $(\overline{alt}\ code)$ -	$(\overline{alt}\ code)$ $(\overline{alt}\ code)$ -
$\frac{1}{2} \Delta C_{max} < \Delta C \leq \Delta C_{max}$	$(alt\ \overline{code})$ $(alt\ \overline{code})$ -	- - -	- - -
$\Delta C > C_{max}$	$(alt\ \overline{code})$ $(alt\ \overline{code})$ -	- - -	- - -

$(\overline{alt}\ code)$ = Potential Code Swap $(\overline{alt}\ code)$

$(alt\ \overline{code})$ = Potential Code Swap $(alt\ \overline{code})$

\- = Use Figure 6-8(a)

ATC-65 (6-8b)

Figure 6-8(b): Association Matrix for Code Swap Reports.

Since only a limited number of associations are permitted for each track, it is important that whenever more than this number are possible the best ones are retained. This implies that some method of scoring association pairs is required. As the geometric zone and level of code and altitude agreement are known for each association, these quantities will be used to construct the score.

No association can ultimately be retained unless some level of altitude agreement exists between the target and the track. Thus, the extent of this agreement is the least valuable scoring discriminant. Experience has shown that neither zone nor identity code agreement is more important than the other; rather, their combination is the key element. These ideas have led to the following scoring formula for an association:

$$\text{score} = (\text{zone-code factor}) \times (\Delta h_{max} + 1) + \Delta h$$

where the zone-code factor is determined as follows:

factor	zone	code
1	1	agree ($\Delta C \leq \Delta C_{max}$)
2	2	agree
3	1	disagree ($\Delta C > \Delta C_{max}$)
4	2	disagree
5	3	agree

Any zone 3 association that failed to agree on code was rejected. Since $\Delta h \leq \Delta h_{max}$ for an acceptable association, the scoring formula gives a different score for each unique association situation.

6.3 Velocity Reasonableness Test

The intent of the Velocity Reasonableness Test is to determine the likelihood of a current target report being part of the same report sequence as that represented by a given track. In order to keep the association process reasonably simple, the association zone boxes have been defined as ρ, θ rectangles centered about the predicted track position. In reality, the locus of possible target positions, as shown in Figure 6-9, is described by a curved surface aligned with the track's velocity vector. Thus, there are some areas of the association box in which target reports should not reasonably appear. The Velocity Reasonableness Test is used to determine when the simplistic box shape has led to unlikely associations being created, so that such associations may be rejected.

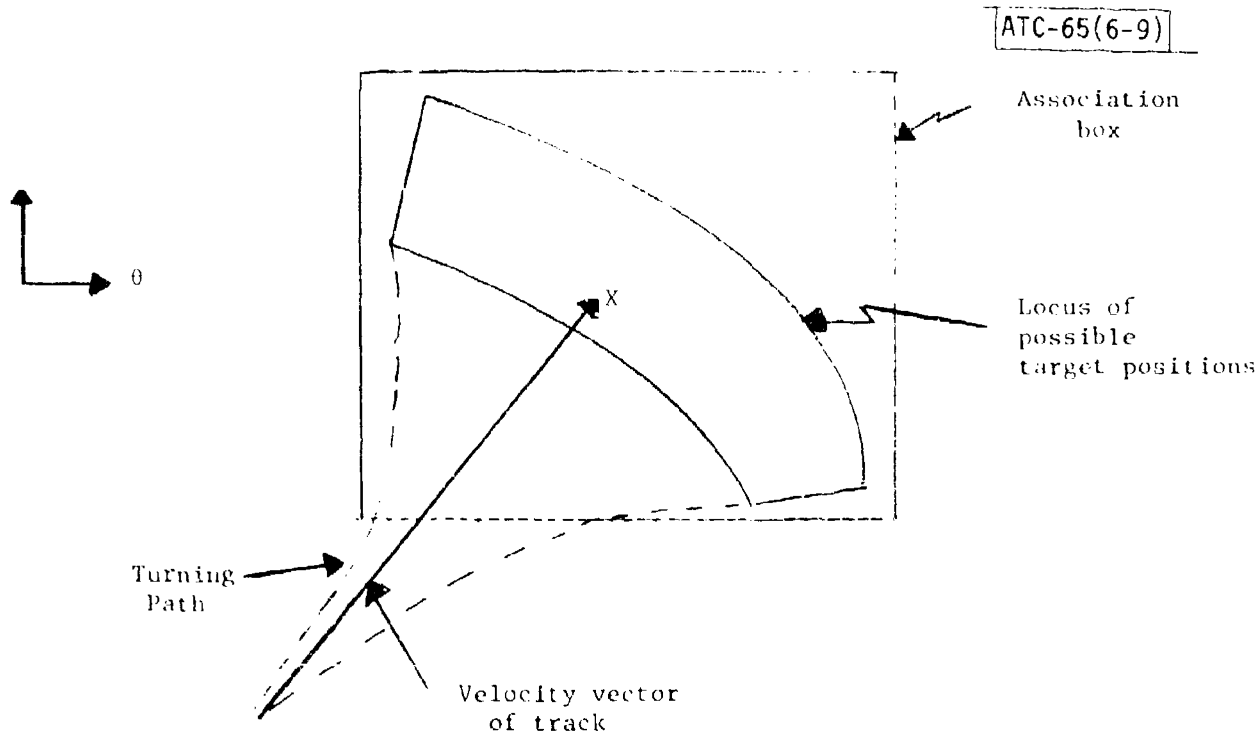


Figure 6-9: Locus of Target Positions

The basic geometry of the test is illustrated in Figure 6-10. Two velocity vectors are employed in the test. The first, which extends from the last known aircraft position to the predicted present position, is the last known velocity for the aircraft under track. The second vector, which extends from the last known aircraft position to the position of the target report in question, would be the actual current velocity of the aircraft if the report in fact corresponds to it. The test basically judges the reasonableness of the required aircraft velocity change.

The coordinate system used for the new and previous velocity vectors, $\vec{v}$ and $\vec{w}$ respectively, depends upon the distance of the track from the sensor. In the normal case, when tracking is being performed in ρ,θ terms, the vector components are slant range and angular distance ($\rho\theta$, not θ, as velocities are being compared). If i and j denote report and predicted track quantities respectively, the two vectors are:

$$\vec{v} = (v_\rho, v_{\rho\theta}) = (\rho_i - \rho_j + \dot{\rho}_j, [\theta_i - \theta_j + \dot{\theta}_j] \times \rho_j)$$

$$\vec{w} = (w_\rho, w_{\rho\theta}) = (\dot{\rho}_j, \dot{\theta}_j \rho_j)$$

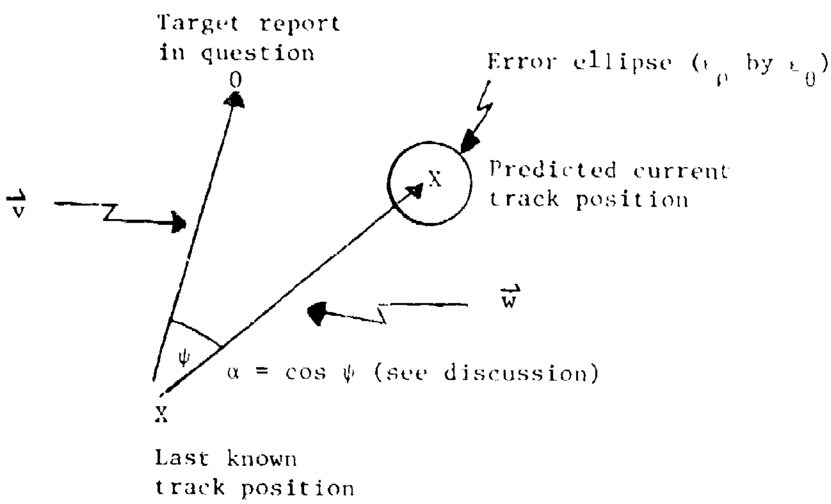
where ρ is in miles, θ in radians, and velocities in per scan units. However, if the track is sufficiently near the sensor that ground x, y tracking is being performed on the track (see section 9.4), these same coordinates are used for the vectors:

$$\vec{v} = (v_x, v_y) = \left(\left[\rho_i^2 - h_j^2\right]^{1/2} \sin\theta_i - x_j + \dot{x}_j, \left[\rho_i^2 - h_j^2\right]^{1/2} \cos\theta_i - y_j + \dot{y}_j\right)$$

$$\vec{w} = (w_x, w_y) = (\dot{x}_j, \dot{y}_j)$$

The vector comparison that constitutes the velocity reasonableness test is accomplished in two parts: angle and magnitude. The direction cosine between the two velocity vectors is given by:

$$\alpha = \frac{\vec{w} \cdot \vec{v}}{|\vec{w}| \times |\vec{v}|}$$



Velocity Reasonableness Test:

expecting vector $\vec{w}$, is vector $\vec{v}$
a likely situation?

Figure 6-10: Velocity Reasonableness Test

The first part of the test is successfully passed if the angle difference is sufficiently small, that is, if,

$$\alpha \geq -(f-1) * P_1$$

where f is the track firmness and P_1 is a parameter. This formula permits up to a 90° angle between vectors for a consistent track (f=1 reduces the equation to $\alpha \geq 0$) and a larger difference if the track is coasting. Thus, a doubling back motion is forbidden for a steady track.

This angle test is not attempted, though, if the measurement uncertainty in either report coordinate is greater than that coordinate's velocity. In such a case, the heading could be in error by more than 90°, thereby invalidating the test. Consequently, the angle test is automatically considered to be passed whenever:

$$\dot{\rho}_j \leq \varepsilon_\rho \ \underline{\text{or}}\ \dot{\theta}_j \leq \varepsilon_\theta$$

for ρ, θ tracks or

$$\dot{x}_j \leq \varepsilon_\rho \ \underline{\text{or}}\ \dot{y}_j \leq \varepsilon_\rho$$

for x, y tracks, where ε_ρ and ε_θ are the system velocity uncertainties.

The magnitude test checks for situations in which the velocity increase exceeds a reasonable limit. The association passes this part of the test whenever:

$$\frac{|\vec{v}|}{|\vec{w}+\vec{\varepsilon}|} \leq f \times P_2$$

where P_2 is another parameter. Again, the test becomes less rigid when the track is coasting. The vector ε is the velocity error vector, given by:

$$\vec{\varepsilon} = (\varepsilon_\rho,\ \rho\ \varepsilon_\theta) \text{ or } (\varepsilon_\rho,\ \varepsilon_\rho)$$

for ρ, θ or x, y systems respectively. Thus, the largest possible $\vec{w}$ is used to be conservative. The dual test, on v being too small, is not made; the angle test partially covers this case, and the error term would lead to automatic success in most situations.

If both component tests are passed successfully, the potential association is acceptable. One possible exception to the use of this test occurs when a close-in x, y track has an unknown altitude. Since such a lack of knowledge severely affects the accuracy of the track ground position and velocity, it may be better to skip the test and accept the situation. This option is a program parameter.

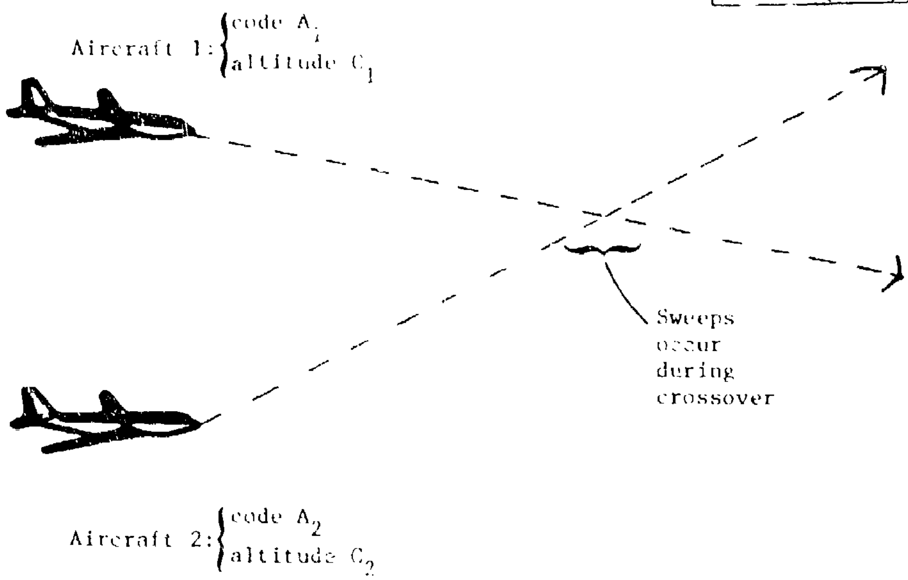
6.4 Code Swapping

Whenever the paths of two aircraft cross each other, it is possible for the situation shown in Figure 6-11 to occur. On the first sweep on which these aircraft respond, say of mode A, the reply from aircraft 1 is received before that from aircraft 2. However, due to differences in transponder delay or other causes, on the subsequent mode C sweep the reply of aircraft 2 is received first. The reply correlation logic, not knowing what the proper pairings should have been, will create the two incorrect target reports shown in this figure. However, since the two reports are very close together, both will be marked as code swap candidates (refer to section 4.6).

When the association process is undertaken for the track corresponding to aircraft 1, both reports 1 and 2 will be identified as candidates for pairings. Report 1, by the rules of the previous section, will agree in code but not in altitude with the track, while report 2 will agree in altitude but not in code. The code swapping procedure described below will then physically interchange the mode A codes of the two reports, creating two reports with the proper mode pairings. The association for report 2 will then be accepted (and highly scored), while the association for report 1 will be rejected.

A target report with an incorrect mode pairing can be created by two other mechanisms in reply correlation. The first situation arises when a fruit reply is received just prior to the real aircraft reply on the first sweep of either mode. Then, as illustrated in Figure 6-12, an incorrect target report and an incomplete target report will result. As before, the track corresponding to the aircraft will find potential code swap associations, one of each type, and initiate a code swapping procedure. Note that the incomplete report may have only one reply; however, since it falls very close in range and azimuth to the other report, a 1-hit report will be created (refer to section 4.6). Had this special 1-hit report rule not have established, the reply would have been rejected as fruit, and the proper code would not have been available for code swapping to use.

Whenever the reply processor makes a high confidence bit error on a reply code, the potential exists for an incorrect target report to result during reply correlation. Figure 6-13 presents a number of example reply sequences for an aircraft that include bit errors, and the target reports that would be created from them. Note that all of the 1-hit reports listed there would be declared by the above-mentioned rule. The last column of the figure indicates how the code swapping mechanism, instigated by the aircraft track, creates a proper target report in all required cases. The general rule for the error correction properties of code swapping can be expressed as follows:



Sweep	Replies received (range ordered)
1	A_1, A_2
2	C_2, C_1
3	A_1, A_2
4	C_2, C_1

Reports Formed:

1. A_1 with C_2
2. A_2 with C_1 — neither agree with either track

After Swapping:

1. A_2 with C_2 - correlates with track 2
2. A_1 with C_1 - correlates with track 1

Figure 6-11: Code Swapping Due to Crossing Aircraft.

Sweep	Replies received (range ordered)	
1	A	
2	C_f,C	(C_f means fruit)
3	A	
4	C	(If this reply is not received, report 2 will contain only 1 reply)

Reports Formed:

1: A with C_f — neither agree with the track

2: - with C

After Swapping:

1: - with C_f - eventually discarded

2: A with C - correlates with track

Figure 6-12: Code Swapping Due to Fruit.

ATC-65 (6-13)

	Replies Received	Reports Produced	After Code Sweeping
1.	A-C-$\overset{*}{A}$-C	A-C	A-C ✓
		$\overset{*}{A}$-	$\overset{*}{A}$-
2.	$\overset{*}{A}$-C-A-C	$\overset{*}{A}$-C	A-C ✓
		A-	$\overset{*}{A}$-
3.	$\overset{*}{A}$-C-A-$\overset{*}{C}$	$\overset{*}{A}$-C	A-C ✓
		A-$\overset{*}{C}$	$\overset{*}{A}$-$\overset{*}{C}$
4.	$\overset{*}{A}$-C-$\overset{**}{A}$-$\overset{*}{C}$-A	$\overset{*}{A}$-C	A-C ✓
		$\overset{**}{A}$-$\overset{*}{C}$	$\overset{**}{A}$-$\overset{*}{C}$
		A-	$\overset{*}{A}$-
5.	A-$\overset{*}{C}$-A-C	A-$\overset{*}{C}$	-$\overset{*}{C}$
		-C	A-C ✓

$*$ = error

$**$ = different error

✓ = report correlated to track

Figure 6-13: Code Swapping Due to Bit Errors.

Whenever at least one reply of each mode (A and C) is decoded properly by the hardware reply processor, a correct target report will exist by the end of track to target association.

The code swapping process is undertaken whenever a track and two of its associating reports satisfy <u>all</u> of the following conditions:

1. The track has no perfect association, which is one in zone 1 or 2 with both code and altitude agreement

2. The two associating reports are both of type potential code swap, one of type (alt code) and the other of type (alt code)

3. Neither of these reports has a perfect association with any other track

4. These reports are spacially close enough together to satisfy the reply correlation range and azimuth conditions

The second condition is necessary and sufficient to insure that code swapping will produce the desired perfect report. The first and third conditions attempt to prevent code swapping when the target reports are due to an aircraft (or aircrafts) different from the one corresponding to the track. The first forbids code swapping when the track already has a perfect report, while the third forbids it when some other track likes one of the reports just the way it is. Note that these two conditions imply that <u>all</u> associations, for all tracks, must be identified before any code swapping can be attempted. This requirement is discussed further in Section 6.6. Finally, the last condition insures that the two reports belong to the same reply correlation ambiguity situation.

When an acceptable code swapping situation is identified, the mode A code and code confidence words of the two target reports are interchanged. The reason for swapping mode A information instead of mode C information, which would appear to be equivalent, is that the former action does not affect the status of any other associations existing for the two reports while the latter action could create new associations or invalidate existing ones. This is because altitude agreement is required for an association while code agreement is not. In addition, if the newly created perfect report shows only 1 reply, the "number of replies" fields for modes A and C are also interchanged between the two reports. This action insures that the good report will be kept in the system while the erroneous reply (due to fruit or bit error) will be eliminated by data editing. Finally, both swapped reports have their mode 2 codes set to the value that results by combining the two individual codes according to the update rules of Section 4.6. This action insures that neither report has an erroneous code (although many low confidence bits will exist), and is the best that can be done due to the absence of mode 2 code in a track file.

6.5 Overall Association Algorithm

The target to track association process commences with two sets of inputs: an ordered list of all tracks currently resident in the sector, prepared by Track Update, and a range sorted list of all target reports to be processed in the sector, prepared by Discrete Correlation. The association procedure processes one by one all tracks not correlated during Discrete Correlation, locating all targets that can be paired with them. A flowchart of all the actions described in this section is provided by Figure 6-14.

The association process for each track begins with the computation of the sizes $\Delta\rho^1$, $\Delta\theta^1$ of its three association zone boxes. These boxes grow whenever a track coasts, as indicated by the formulas derived in Section 6.2. Then, the range interval in which associating reports must lie for track j is given by:

$$\rho_j - \Delta\rho^3 \leq \rho \leq \rho_j + \Delta\rho^3$$

The set of targets to be considered for association are all those residing in any range sort bin contained wholly or partially within this interval. Targets already correlated during Discrete Correlation are ignored, while all others encountered in these bins are processed through the set of tests described below. The single exception to this rule is that should 1-hit reports be generally permitted, due to a very low fruit environment, they may not associate with non-established tracks (i.e.: those with 5 or fewer reports). As explained in Section 10.3, this prevents the continuation of extraneous tracks. 1-hit reports created for code swapping, however, are exempt from this restriction.

The entire set of tests can be bypassed if the target report under consideration was carried over from the previous sector, as the result can be obtained by consulting information left in the association cross reference table from that sector. If the track was processed in the previous sector, its new linked list sequence number is guaranteed to be no larger than its sequence number in the previous sector. This is because carried over tracks are placed in order at the head of the next sector's list (see Section 9.6). Thus, the track's current number equals the old one if all tracks before it were also carried over and is smaller otherwise. This condition insures that the previous sector association information for the track cannot yet have been overwritten.

If the current target appears on the track's last sector list, the score is simply copied; if the target fails to appear, it is known that the association was rejected. Finally, if the track did not exist in the previous sector, the association with the target can be rejected. This follows from the fact that if the track had wished to associate with reports from the previous sector, it would have been in that sector looking for them.

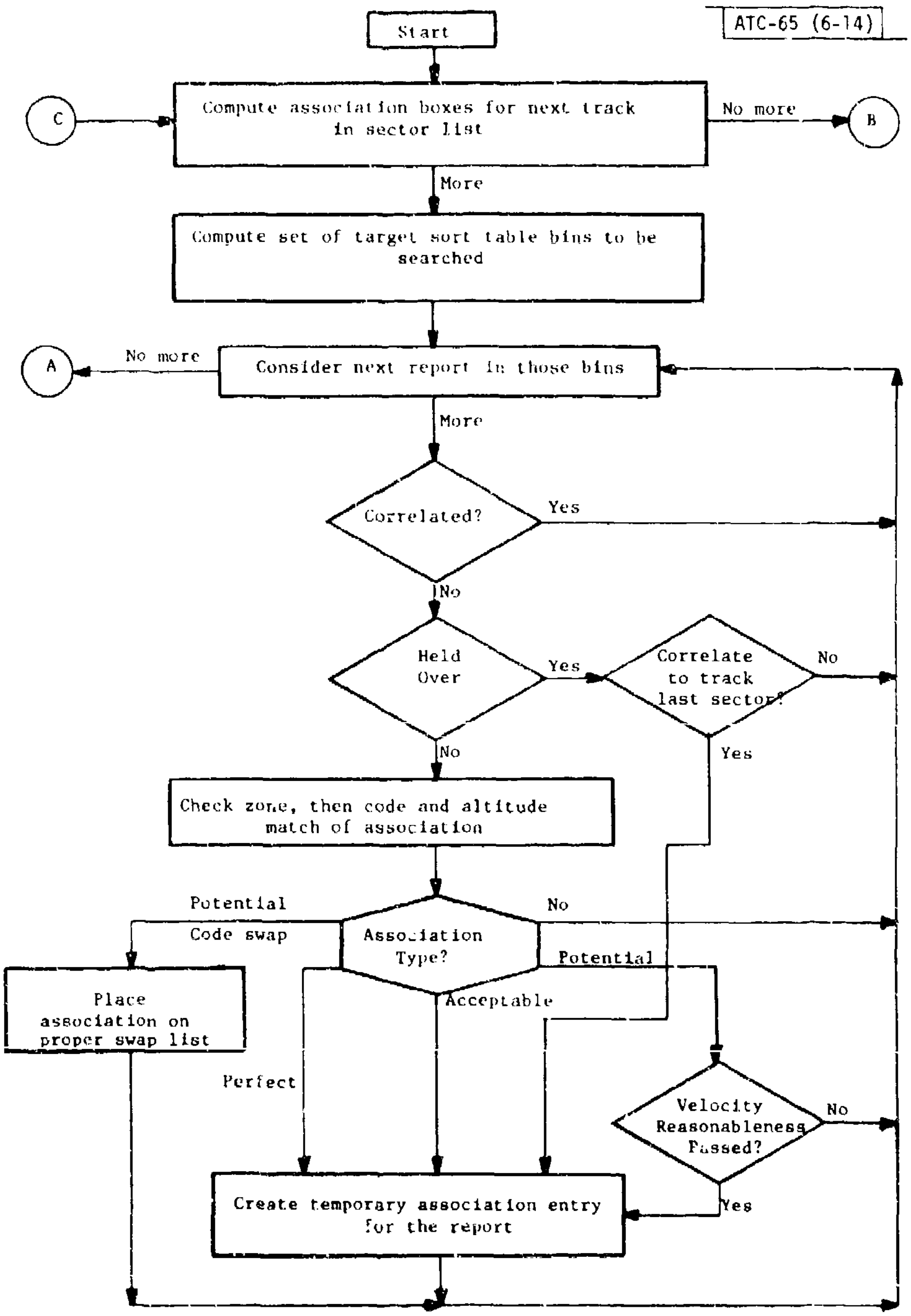


Figure 6-14: Association Flowchart (1 of 3)

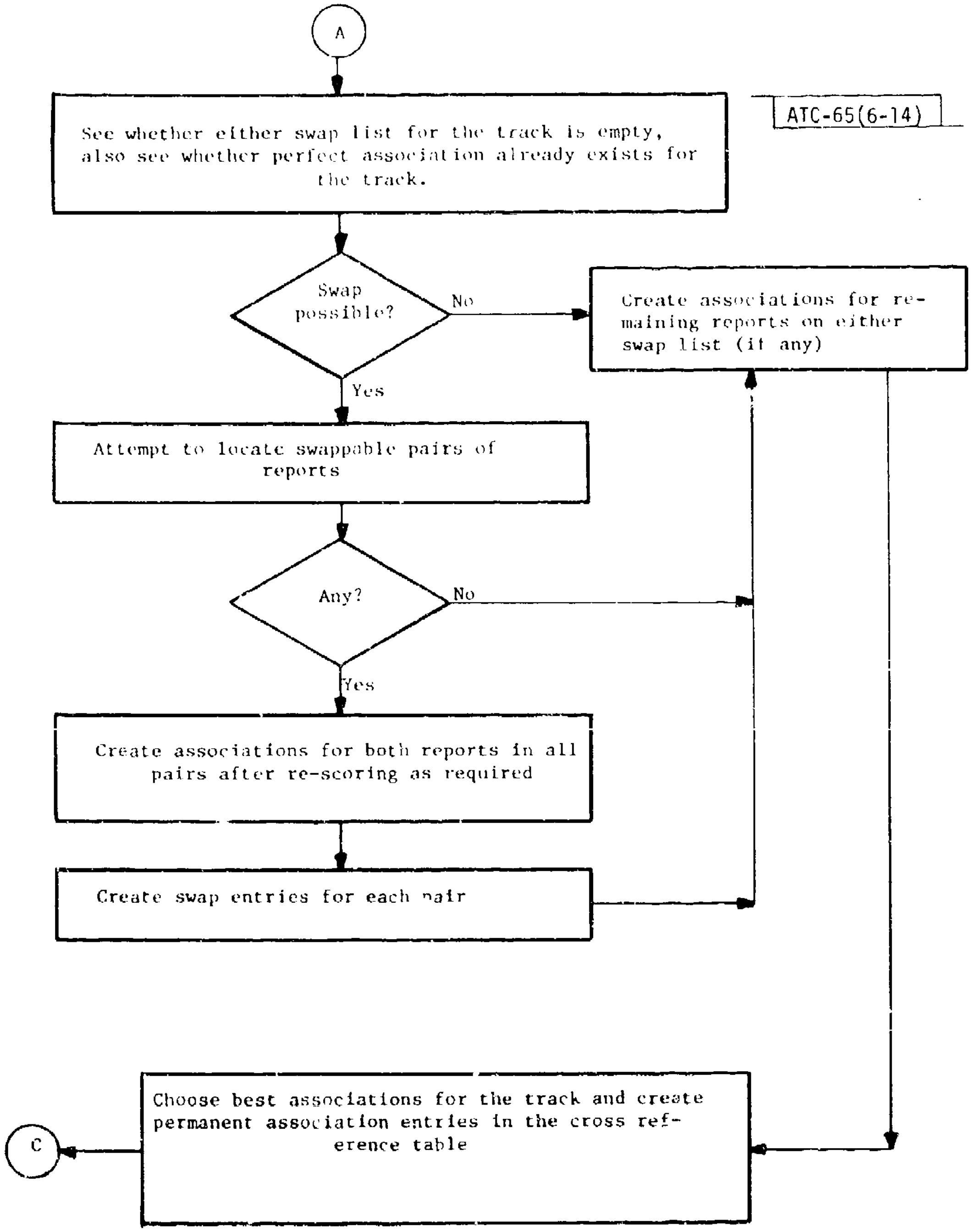


Figure 6-14: Association Flowchart (2 of 3)

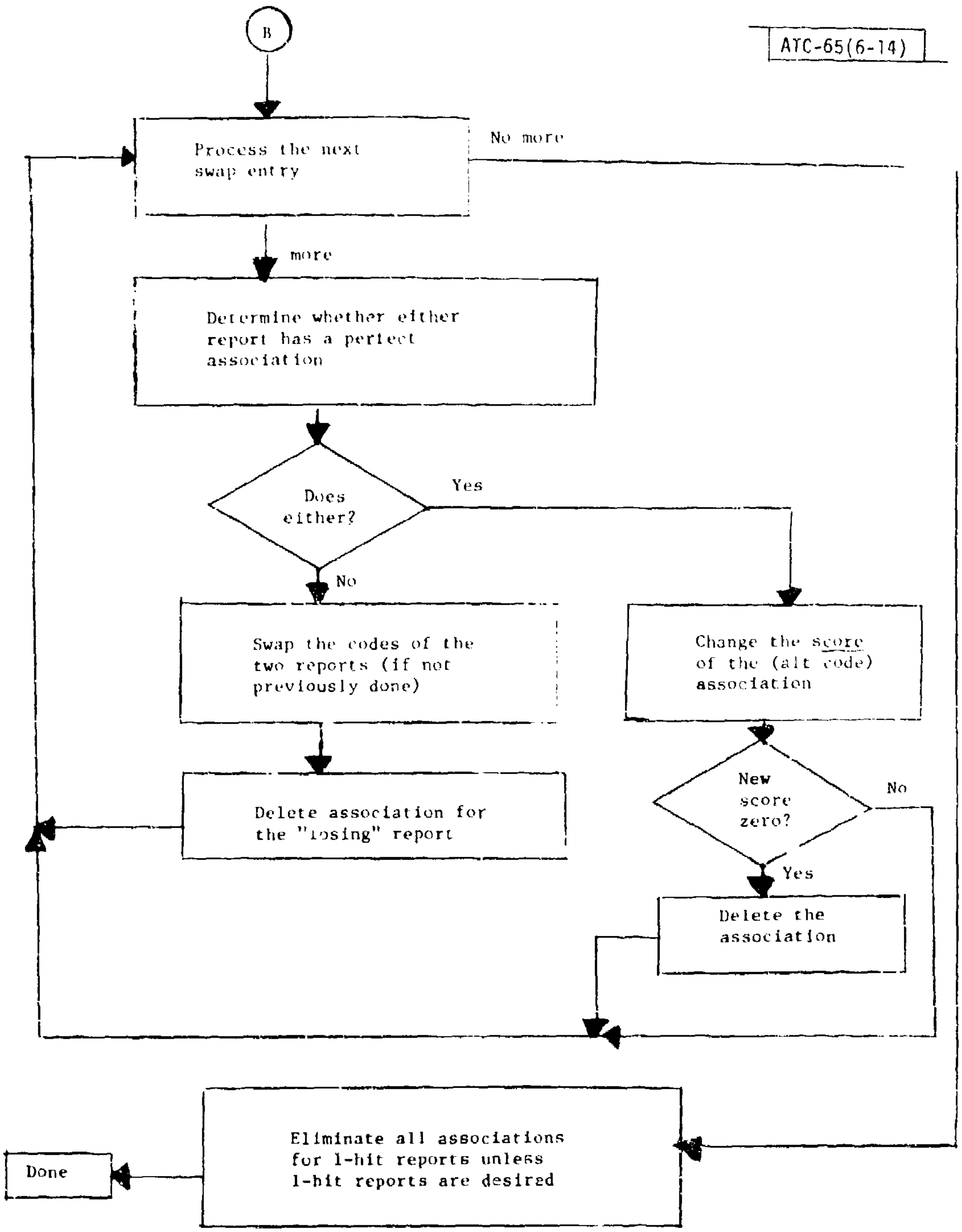


Figure 6-14: Association Flowchart (3 of 3).

When the report is a new one, the association process commences with the determination of the association zone by the procedure presented in section 6.2. If the report falls outside of the track's zone 3 box, it is rejected immediately. Otherwise, the zone of the association is recorded and code and altitude checking proceed as described in section 6.2. When these tests are completed, the association type is determined from the appropriate matrix of Figure 6-8.

If the type is no association, the report is rejected and the next one is processed. If the type is potential association, the Velocity Reasonableness Test is performed. Should the target/track pair fail the test, the association is rejected; otherwise, the association is converted to acceptable. A perfect or acceptable association is scored according to the rules described in section 6.2, and a temporary association entry containing the report number and the score is created for it.

Potential code swap associations cannot be fully resolved at the time they are created. Instead, any (alt $\overline{\text{code}}$) association is placed on one swap list, and any ($\overline{\text{alt}}$ code) association on a second list. Each such entry contains the target number and the score the association would receive if no code swap using it were to occur. This value is obtained by referencing the association type corresponding to its attributes as given in Figure 6-8(a). If the result of this check is a rejection, a score of zero is used.

After all possible associations for a track have been created, the two swap lists are processed (if non-empty). Although actual code swapping cannot occur until all tracks have been processed, much of the preliminary work can be accomplished at this time. First, if the track has any perfect associations, no code swapping initiated by it is possible. Thus, in such a case, all associations on the code swap lists can be entered onto the temporary association list using the scores already determined for them, except that those whose scores are zero are rejected. In addition, the same action can be taken if either swap list is empty.

If however, code swapping cannot be ruled out, an attempt is made to locate pairs of swappable reports. That is, all (alt $\overline{\text{code}}$) list reports are compared with all ($\overline{\text{alt}}$ code) list ones to find pairs that satisfy the range and azimuth correlation conditions. Each such pair is processed in the manner specified below. All associations that remain on either list after the swap pairs are identified are entered onto the temporary association list (or rejected) as described above.

Although the large majority of all potential code swaps will in fact be consummated, no guarantee can be given during individual track processing. Thus, both eventualities must be covered. The method that accomplishes this aim is the following. First, the (alt $\overline{\text{code}}$) association of each swap pair, which is the one that would become perfect after swapping, is rescored under

the assumption $\Delta C = 0$, the after swap value. Using this score, the association is placed on the temporary association list. Next, the ($\overline{\text{alt}}$ code) association is placed on this list with the score previously calculated for it, which assumed no swap would occur. Finally, a swap entry is created for the pair on the sector swap list. Such an entry, as depicted in Figure 6-15, contains four fields: the track initiating the swap, the (alt $\overline{\text{code}}$) report of the swap pair, the ($\overline{\text{alt}}$ code) report, and the originally computed score for the (alt $\overline{\text{code}}$) association.

This procedure guarantees that the (alt $\overline{\text{code}}$) association, which probably will become perfect, is scored very highly and thus will be one of the ones retained. On the other hand, the ($\overline{\text{alt}}$ code) association is maintained just in case the swap should be prevented by another track. Should the code swap later occur as expected, the former association will have the proper score, while the latter one can be deleted at that time. However, should the swap be blocked, the ($\overline{\text{alt}}$ code) association will be properly scored, while the proper score to substitute for the (alt $\overline{\text{code}}$) one is contained within the swap list entry. If this value is zero, of course, the association must be deleted. A pictorial summary of the actions that occur when a code swap is and is not permitted is presented in Figure 6-16.

After the partial code swap resolution is completed for a track, the number of temporary associations created is compared with the maximum permissable number. If acceptable, all of them are converted to permanent form through creation of an entry in the cross reference table (refer to section 6.1). If too many associations exist, however, those with the lowest (best) scores are chosen, while all others are discarded. In addition, should both associations of any swap pair be eliminated by this pruning action, the swap list entry corresponding to it must be deleted.

Finally, after all tracks in the sector have progressed through the association process, the actual code swapping actions are performed. The swap list, if non-empty, is processed one entry at a time. If neither report in an entry has created a perfect association with any track, the code swap itself is carried out as described in section 6.4; otherwise, the code swap must be ignored. In either event, the association status of the two relevant associations for the initiating track are adjusted in the manner described above. Should the same pair of reports exist in two different swap entries, such as would occur when the reply correlation error being corrected was caused by two crossing aircraft, the codes are swapped only once.

In addition, since 1-hit swap candidate reports were created solely for use in this code swapping process, they must be removed from the system at this point (if one was made into a perfect match through a code swap, the other report of the pair becomes the 1-hit report as was stated in section 6.4). Any associations they may have formed must also be dropped. This set of actions is not taken, however, if the 1-hit report option is to be employed for the sensor due to very low fruit rates.

ATC-65 (6-15)

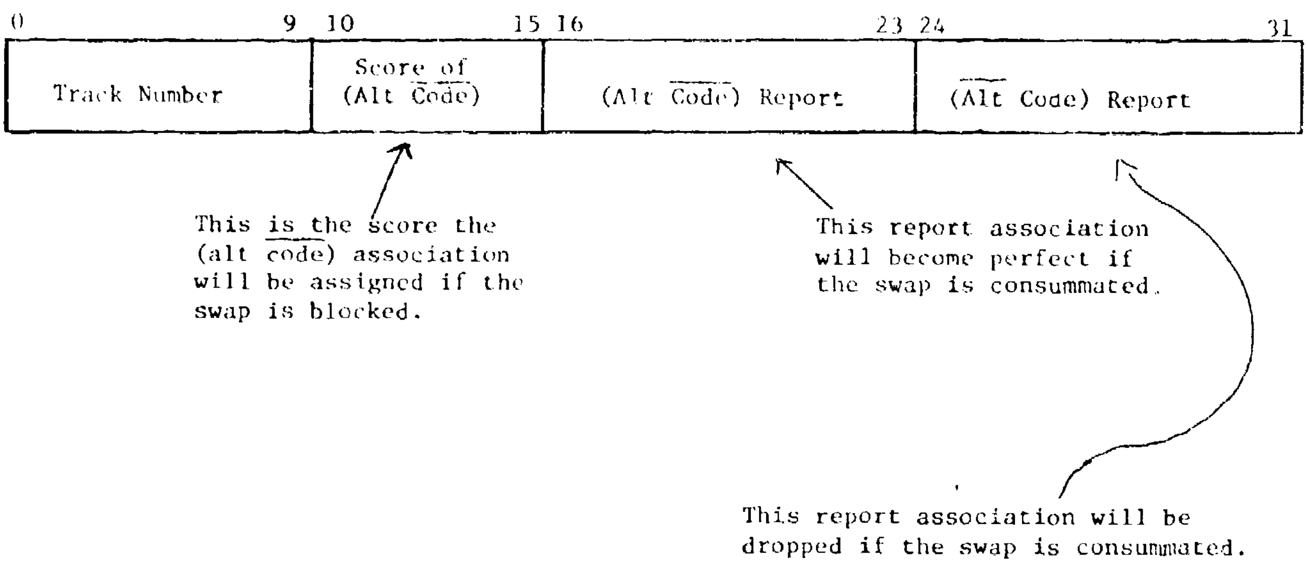


Figure 6-15: Swap List Entry

Before Swap Attempted:

Report i (alt code)	Score S_i
Report j (alt code)	Score S_j

Temporary association entries for reports of the swap pair for track k

Track k	S'_i	Report i	Report j

Swap list entry for these reports

If Swap Succeeds:

Report i (perfect)	Score S_i

Report j association is dropped

If Swap Fails:

Report i	Score S'_i
Report j	Score S_j

Report i association has the score from the swap list entry subsituted for it

Figure 6-16: Swap Resolution Actions.

7.0 TARGET TO TRACK CORRELATION

Once all associations for each track have been determined, it becomes possible to decide which target report, if any, should be used to update each track file. In virtually all cases, only one report will associate with a given track, and that report will associate with no other track. In such situations the selection is obvious. In most other cases, two or more reports will associate with one track, or two or more tracks with one report. In these situations the "best" association is chosen. The ranking of the associations is accomplished through use of either the Quality Score or Deviation Score of the corresponding track/target pair. The Quality Score measures differences between the track and target attributes, while the Deviation Score, employed only when Quality Score ties exist, measures and weights the geometric difference between the track prediction and report positions. Occasionally several tracks and several reports will associate with each other. These situations are resolved by selecting the set of target/track pairs that minimizes the total system Quality Score.

In all cases, a target to track correlation is accepted only if the track is ready to correlate. If, on the other hand, the track can reasonably expect to find a superior report in a subsequent sector, the correlation is postponed. Both the track and target report are then carried over into the next sector, where the association process is again performed.

7.1 Quality Score

The Quality Score of a track/target association pair is a measure of the relative differences between their attributes and of the degree of certainty that each entity represents a real aircraft. The following components are incorporated into the Quality Score. Since most were already determined during the association process, little extra computational cost is attached to the scoring mechanism.

1. Mode A code agreement
2. Association zone
3. Number of replies in report
4. Altitude agreement
5. Track confidence

Figure 7-1 presents in detail the manner in which each of these items is evaluated as well as the individual scores for each possible result. The final Quality Score for the association, as indicated in the figure, is the octal concatenation of the component test scores.

Octal Digit and Factor	Condition	Score
7 (most significant) zone-code	zone = 1, code agree	0
	zone = 2, code agree	1
	zone = 1, code disagree	2
	zone = 2, code disagree	3
	zone = 3, code agree	4
6 number of replies (modes A and C only)	3 or more	0
	2 of same mode	0
	1 of each mode	1
	1 reply	2
5 code agreement	$\Delta C = 0$, all bits high confidence	0
	$\Delta C = 0$, some bits low	1
	$\Delta C = 1/2\ \Delta C_{max}$	2
	$\Delta C = \Delta C_{max}$	3
	$\Delta C = 2\Delta C_{max}$ and:	
	some bits low, track code in transition	4
	all bits high, track in transition	5
	some bits low, track steady	6
	all bits high, track steady	7
4 altitude agreement	$\Delta h \leq 500$ feet	0
	$\Delta h = 600$ feet	1
	$\Delta h = 700$ feet	2
	$\Delta h = 800$ feet	3
	$\Delta h = 900$ feet	4
	$\Delta h = 1000$ feet	5
	$\Delta h > 1000$ feet	6
3 track validity	track established, $\rho \geq \rho_v$	0
	track established, $\rho < \rho_v$	1
	new track, $\rho \geq \rho_v$	2
	new track, $\rho < \rho_v$	3
2, 1, 0 deviation score		

$$\text{Quality Score} = (d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0)_8$$

Figure 7-1: Quality Score Determination

Since it is impossible for the score of one component test to "spill over" into the digit of the next one, this Quality Score is actually an implementation of a multi-stage decision algorithm. That is, if two associations exist for a track, the one chosen will be the one with the lower code-zone (digit 7) score, even if that association lost on all other criteria. If the associations tie on this criterion, however, the decision will be based on the next item, etc. Because all the decision item scores are combined into a single number, however, a single comparison will automatically implement the entire test hierarchy, selecting the winning association on the basis of the first non-tied decision stage.

The value of the first test, association zone and gross mode A code agreement, can be determined directly from the score of the association. By referencing Section 6.2, it is seen that

$$\text{value of digit 7} = \frac{\text{association score}}{\Delta h_{max} + 1}$$

where integer division (no remainder) is employed.

The second component, number of replies of modes A and C constituting the target report, can be obtained directly from the corresponding fields of the target report. The reason for penalizing a target with one reply of each mode is that such a reply grouping is characteristic of a report formed by coincident fruit. Fruit of the same mode would require code agreement to correlate, and thus most reports with two replies of the same mode are real. Although 1-hit reports are generally not permitted in the system, they may be employed by sensors in very low fruit environments.

The third component test is a finer measure of code agreement between target and track than that employed in the first test. As expected, the best score (lowest number) is given when all code bits agree and are declared with high confidence, while the worst is obtained when the codes disagree in several high confidence bit positions, the target code is all high confidence, and the track code is not in transition, meaning that its last correlating report has confirmed its code. Code disagreement is not penalized as severely when uncertainty exists in the target code as bit decisions are often made incorrectly when garble is present. Similarly, if the track code is in transition, less weight is given to code disagreement. The case of code agreement with $\Delta C = \Delta C_{max}$ exists when the track and report codes differ by no more than a parametric number of bits (typically one), and thus this situation falls between agreement and disagreement. The elements required for this test are obtained as follows: code agreement or disagreement is defined as for the first test, ΔC is computed as defined in Section 6.2, the degree of target code uncertainty is determined by examining the report code confidence field (all 0's = high confidence, all 1's = unknown, mixed 1's and 0's = some uncertainty), and the track transition count is part of the track information ensemble.

The next test measures the amount of difference between the track and target altitudes. Note that altitude differences of greater than Δh_{max}, normally 1000 feet, would have prevented association from occuring in the first place. Thus "disagreement" is not possible, which explains why this seemingly important test ranks so low in the hierarchy. The value of Δh in hundreds of feet has already been computed during association and resides in the association score Thus, it can be obtained as:

$$\Delta h = 100 \times [\text{association score} - (\Delta h_{max} + 1) \times (\text{value of digit 7})]$$

which follows from the definition of association score given in 6.2 and the digit 7 discussion above.

The final component of the Quality Score gives an edge to tracks that are likely to correspond accurately to real aircraft positions. Thus, tracks which have successfully correlated a number of times are rated better than newly initiated ones, while tracks passing over the sensor, where positional prediction accuracy is often hurt by missing data or uncertain altitudes, are rated below more distant tracks. The former rule also has the advantage of reducing track drops during splits. For example, assume reply correlation generates two reports for an aircraft on two successive scans. These reports will initiate a new track which will compete with the original one. By giving priority to the established track, assurance is provided that this track will be the one to continue correlation after the split cause has disappeared.

The final three octal digits of the Quality Score are reserved for the Deviation Score when its calculation is required. This permits the total score to be represented as one entity.

It should be noted that the component tests of the Quality Score could be reordered in any manner. The present level of experience with real data, however, seems to indicate that the hierarchy described here is proper.

7.2 Deviation Score

It is quite possible that the Quality Scores of two associations will be identical. For example, reports from two general aviation aircraft, both reporting a code of 1200 and having no encoding altimeters, would often produce the same score relative to any track. The intent of the Deviation Score is to break such ties by taking into account the geometric difference between the track and target positions.

The Deviation Score doesn't merely reflect the distance between the positions; rather it indicates the likelihood of the aircraft under track being at the position represented by the target report. In particular, the scoring rules employ the fact that changes in aircraft speed from scan to scan are unlikely, most changes in aircraft velocity being caused by turns.

When an aircraft executes a turn of unknown magnitude, the set of possible locations it can reach, as shown in Figure 7-2, is defined by a region which is fairly wide in the crosstrack direction and narrow in the along-track direction. Since the association box constructed about the track position, also illustrated in the figure, must be square and in ρ, θ rather than track oriented coordinates to prevent excessive computation time, it includes much area quite distant from this region. By using the track-oriented deviation zone, otherwise unresolvable multiple association cases can be solved easily. For example, although the two tracks of Figure 7-3 are predicted to the same spot, the report that belongs to each track is decided easily through the deviation boxes.

The Deviation Score represents an approximation to these ideas. As depicted in Figure 7-4, the accessible region for the aircraft is represented as a rectangle and the turning locus as two line segments. The score assigned to each point in this region is then computed as the product of two factors: one that penalizes absolute distance from the predicted position and the second that penalizes deviations from the turning locus. The two vectors needed for this computation, as shown in Figure 7-5, are:

$$\vec{d} = (\Delta\rho,\ \rho\Delta\theta)$$

$$\vec{t} = (t_\rho,\ t_\theta)$$

The former represents the deviation of the report relative to the predicted track position, while the latter is a unit vector in the direction of the turning locus. The actual computation formulas for the $\vec{t}$ components are supplied by the figure.

The penalty factor for absolute distance between target and track is defined to be:

$$f_1 = \left|\frac{\Delta\rho}{\varepsilon_\rho}\right| + \left|\frac{\rho\Delta\theta}{\rho\varepsilon_\theta}\right|$$

where ε_ρ and ε_θ are the 3σ report measurement errors. The factor that rates the direction of this deviation does so by comparing its components in the directions parallel and perpendicular to the turning locus. That is:

$$C_{par} = \vec{d} \cdot \vec{t}$$

$$C_{perp} = \sqrt{|\vec{d}| - C_{par}^2}$$

$$f_2 = \frac{C_{perp}}{C_{par}} \qquad 0.5 \leq f \leq 2$$

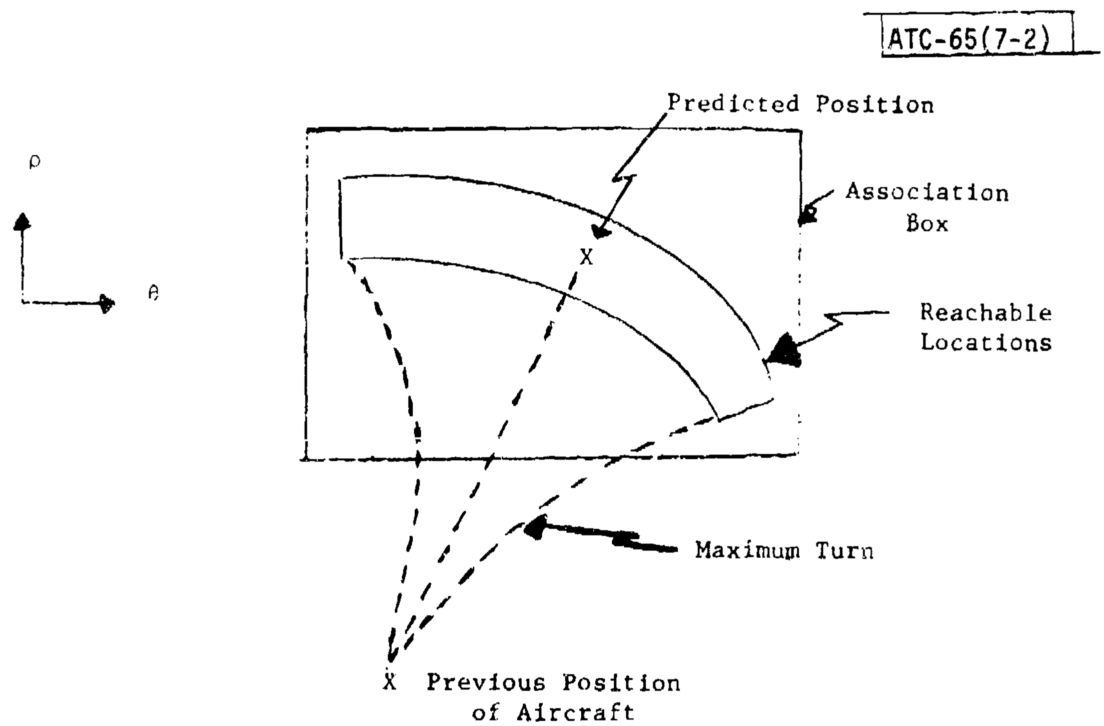
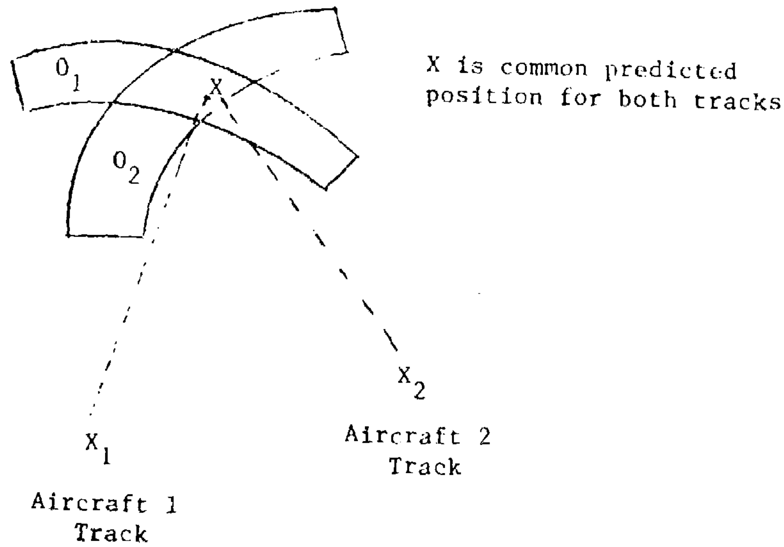


Figure 7-2: Reachable Aircraft Locations.



Resolution is seen to be:

Track 1 with Report 1

Track 2 with Report 2

ATC-65(7-3)

Figure 7-3: Resolving Ambiguities through Deviation Zones.

AIC-65(7-4)

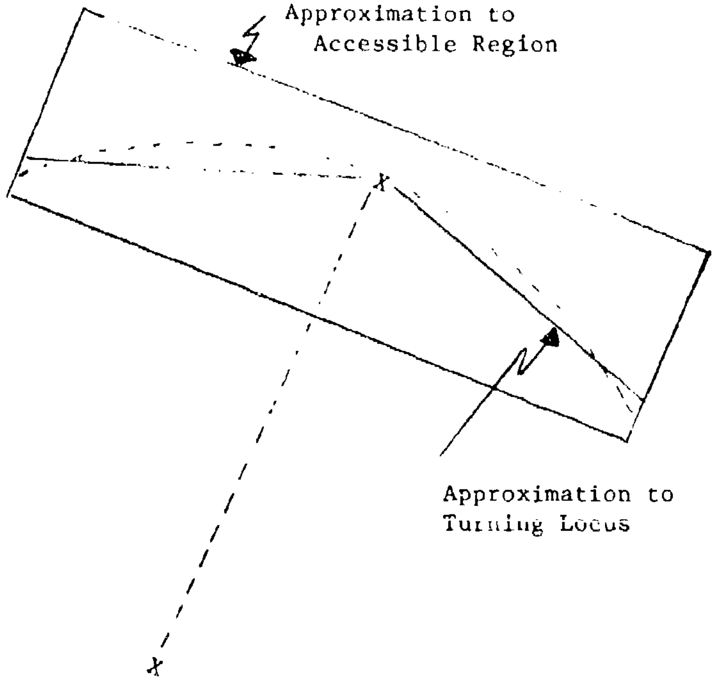
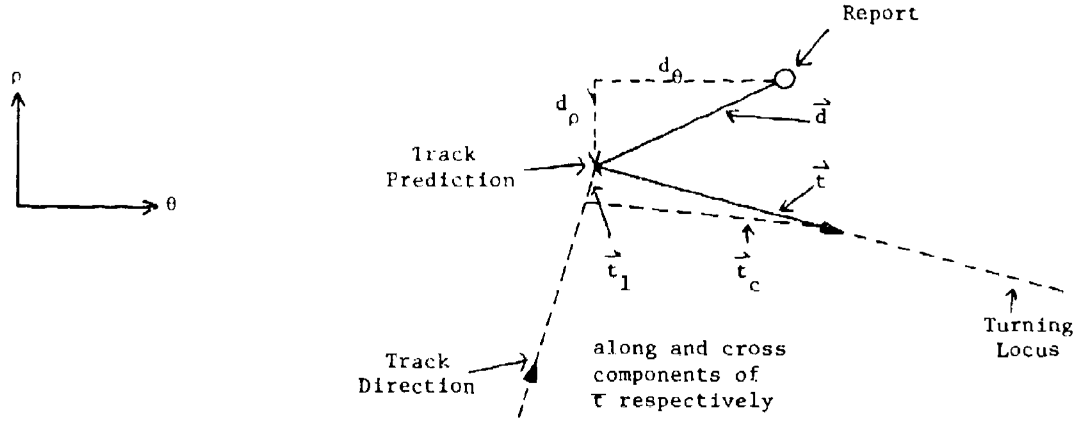


Figure 7-4: Approximations for Deviation Region.



assume turn rate of w radians per scan

Let:

$$A = \frac{1 - \cos w}{w}$$

$$B = 1 - \frac{\sin w}{w}$$

$\vec{d} = (d_\rho, d_\theta) = (\Delta\rho, \rho\Delta\theta)$ $\quad\quad$ Δ = report-track

$\vec{t} = \vec{t}_c - \vec{t}_1$

$\vec{t}_c = (t_{c\rho}, t_{c\theta}) = (-A\rho\dot{\theta}, A\dot{\rho})$ $\quad\quad$ track coordinates and velocities

$\vec{t}_1 = (t_{1\rho}, t_{1\theta}) = (B\dot{\rho}, B\rho\dot{\theta})$

For X-Y coordinates, make following substitutions:

X for ρ

$\dot{X}$ for $\dot{\rho}$

Y for $\rho\theta$

$\dot{Y}$ for $\rho\dot{\theta}$

Figure 7-5: Deviation Score Vectors

Thus, deviations due to turns ($C_{perp} \simeq 0$) are penalized very little compared to those requiring along-track accelerations. The bounds on f_2 prevent its effect from overshadowing that of f_1. The final deviation score is given by:

$$D = f_1 \times f_2$$

This score is quantized to $25^{\underline{ths}}$ and added to the Quality Score in the manner indicated above.

For tracks near the sensor, x,y coordinates are employed instead of ρ, θ ones (refer to Chapter 9). For this situation, both the $\vec{d}$ and $\vec{t}$ vectors are computed with x, y components:

$$\vec{d} = (\Delta x, \Delta y)$$

$$\vec{t} = (t_x, t_y)$$

where Figure 7-5 gives the latter component equations. Also, the first deviation factor is expressed as:

$$f_1 = \left|\frac{\Delta x}{\varepsilon_\rho}\right| + \left|\frac{\Delta y}{\varepsilon_\rho}\right|$$

7.3 Correlation Timing

If the association boxes for all tracks were contained within a single sector, the association and correlation processes could both be performed during that sector and timing would never be a problem. However, whenever a predicted track position occurs near a sector boundary azimuth, it is possible that the association box for the track will encompass two or more sectors. In fact, if the track is very near the sensor, its box could include parts of every sector. Clearly, if every possible associating target is required before correlation can occur, the correlation decision might be delayed several seconds. In the worst case, when many tracks and many targets associate with each other, no closed system might ever occur, and hence no correlation decision could be made. Since target reports are required to be processed as soon as possible, and no delay exceeding a parametric number of sectors is permitted, a compromise correlation procedure is required.

The design implemented to handle this issue is the following. Define MS to be the maximum number of sectors for which correlation of a target may be delayed. Also define BS and LS to be the number of sectors prior to and following the center sector respectively over which the track's correlation box extends. Then the track begins to seek associating targets ER = Min{MS, BS} sectors before its predicted sector. The track will not be permitted to correlate, though, before targets from its predicted sector have been received,

as that is where the correct target is most likely to occur. This rule explains why a track will never be permitted to associate with targets earlier than MS sectors before its predicted one; by the time the track was allowed to correlate, these targets would have already been output.

Once the targets from the predicted sector have been received, correlation for the track will be attempted. If a correlating target is identified, the correlation will be accepted provided at least one of the following three conditions is met:

1. The first octal digit of the Quality Score of the association is lower than a specified value (and thus the association is good enough to justify ending the search).

2. The target has already been delayed for MS sectors and thus cannot be held any longer in the system.

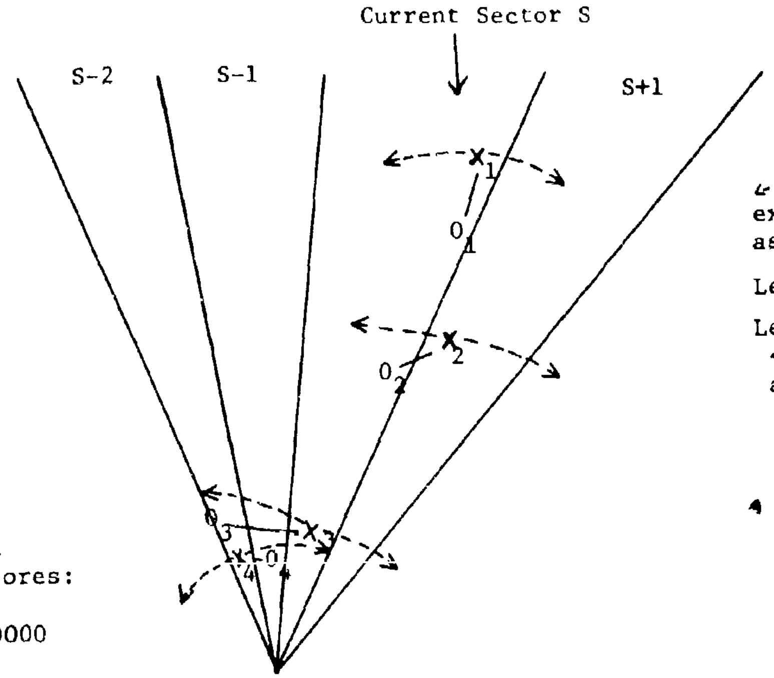
3. The track has already received targets from the sector LR = Min(MS, LS) later than its predicted one (and thus has completed its search).

If none of these conditions is satisfied, correlation is postponed for another sector. Thus, in a many-track-many-target association system, correlation will be performed as specified in Section 7.5 if any of the tracks or targets requests it. Some of the resulting correlation pairs may be accepted due to satisfaction of one of these rules, others may be rejected because no rule was satisfied, and still others may be rejected because the system had not yet received targets from the predicted sector of the track. Figure 7-6 illustrates the resolution process for a typical situation.

The method used to provide the data required for these decisions is the assignment of an integer flag to each target report and each track. Figure 7-7 presents the interpretations assigned to the various values the track flag TF can assume. When a track is updated, the first sector in which it will seek associations is determined as described above and its initial flag value is set accordingly (see Section 9.6 for a detailed discussion). As long as the track's flag is non-zero at the end of the correlation process for a sector, track update will merely recompute the flag value (if necessary) and move the track to the list for the next sector. When the flag has been set to zero by correlation, indicating a successful correlation or a coast condition, the track is updated to the next scan and the process starts anew.

The target flag rules, also shown in Figure 7-7, are considerably simpler. A target is assigned a flag of zero when it is created. If it must be delayed in the system due to its becoming associated with a track that is moving to the next sector, its flag is set to indicate the last sector in which it can be processed. The report can then by delayed further, if required, but not beyond this final sector. Note that even close-in reports are not delayed at all unless a track associating with them requests it. Any track in the system that wished to associate with this target would be included in the list for its sector, and thus delaying the report cannot lead to later associations.

ATC-65(7-6)



extent of track association box

Let MS = 2

Let Quality Score < 20000000 be acceptable

Association Quality Scores:

Q_{11}: 10000000

Q_{22}: 10000000

Q_{33}: 40000000

Q_{44}: 30000000

Correlation Pair	Accepted?	Reason
$X_1 - 0_1$	Yes	Score acceptable
$X_2 - 0_2$	No	Track centered in subsequent sector
$X_3 - 0_3$	Yes	Report cannot be delayed further
$X_4 - 0_4$	Yes	Track has reached end of search

Figure 7-6: Correlation Timing Example

Track Flag (TF)	Interpretation
$TF = 0$	Update track this sector.
$10 < TF \leq 74$	Track's association box is centered in subsequent sector TF - 10.
$74 < TF \leq 138$	Center of track's association box has already been reached; box ends at subsequent sector TF - 74.
$TF = 139$	End of track's association box has already been reached; correlate as soon as possible.

Target Flag (GF)	Interpretation
$GF = 0$	Process target this sector.
$GF > 0$	Target delayed, but no further than sector GF is permitted.

Figure 7-7: Timing Flag Values

When a correlation has been selected, and the timing considerations permit it to be consummated, the actual actions performed are the following:

1. The target number is placed into the proper field of the track file entry.

2. The track number is placed into the proper field of the target report.

3. The track flag is set to zero.

4. The target flag is set to zero.

7.4 Elementary Correlation Cases

There are three association situations in which the selection of the proper target/track pair to correlate is straightforward. These cases are the following:

1. One target and one track associate only with each other (1 on 1)

2. One target associates with many tracks, but each track associates only with that target (m on 1)

3. One track associates with many targets, but each target associates only with that track (1 on n)

Once the proper pair is chosen, the correlation is actually performed only if the timing criteria of the previous section are satisfied. Figure 7-8 presents a flowchart of the algorithm for these cases.

For the 1 on 1 case, which is by far the most common, no Quality Score is required if the track is in its last correlation sector or if the report cannot be delayed any longer. In other situations, only the first digit of the Quality Score is required to determine whether correlation can be consummated. Since this digit is contained within the association score (refer to Section 7.1), again no processing is required. Thus, the usual correlation case introduces little execution overhead.

When either of the many to one (m on 1 or 1 on n) association situations arises, correlation is attempted if any of the tracks or reports are ready. First, the Quality Scores for all associations are computed in full. Then the lowest score is identified. If there is a tie for the best score, the Deviation Scores for the tied associations are evaluated and added to the Quality Scores. Should a tie still exist, which is rare, random selection is employed.

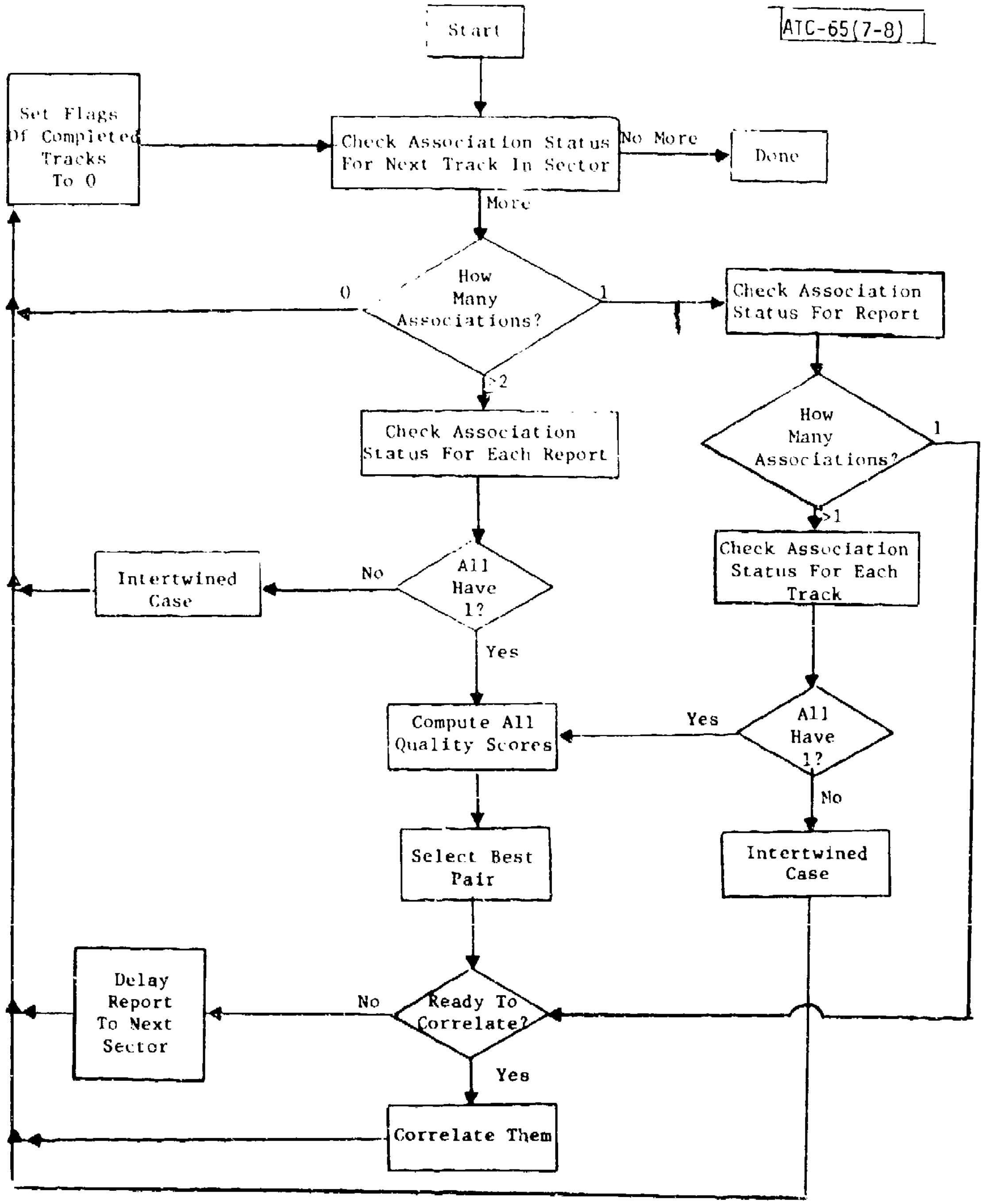


Figure 7-8: Elementary Correlation Logic

Once the target/track pair to correlate is identified, the timing criteria of Section 7.3 are checked to determine whether or not the correlation is acceptable. If it is, correlation is performed; if not, both the track and target are carried over to the next sector through the flagging mechanisms described in the last section. Leftover tracks in n on 1 situations that have reached their last correlation sector, that is those whose flags equal 139, have their flags set to 0 to indicate they should be coasted; other leftover tracks have their flags left unchanged so that they can attempt correlation in the next sector. All leftover reports in 1 on m situations have their flags set to 0, which will result in their being treated as uncorrelated reports. There is no reason to bring any of them into the next sector since any track whose association box included their positions would have been present in the current sector.

7.5 Intertwined Correlation Cases

Selecting the proper correlation pairs becomes considerably more difficult when the association situation consists of m tracks and n reports associating with each other. Although frequently each track can be assigned its first choice report, there is no guarantee that conflicts will not result. Thus, some objective function must be defined in order to be able to decide when one set of correlation pairs in superior to another. The function that has been selected is the minimization of the sum of the Quality Scores for the pairings chosen, where each uncorrelated track or report is assigned a penalty Quality Score.

Mathematically, this function can be expressed as follows. Define

$$X_{ij} = \begin{cases} 1 \text{ if track i associated with report j} \\ 0 \text{ otherwise} \end{cases}$$

$$X_{i,\, n+1} = 1 \text{ for all } i$$

$$X_{m+1,\, j} = 1 \text{ for all } j$$

where correlation with track m + 1 (or report n + 1) will be used to indicate an uncorrelated report (or track). The Quality Score for each real association ($X_{ij} = 1$, $i \leq m$, $j \leq n$) is given by the rules of Section 7.1, while that for each auxiliary association ($i = m+1$ or $j = n+1$) is assigned the default value, currently set at octal 50000000. Next define

$$Y_{ij} = \begin{cases} 1 \text{ if track i paired with report j} \\ 0 \text{ otherwise} \end{cases}$$

as the correlation pair assignment variables. Then the optimum correlation resolution is described as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} Y_{ij} Q_{ij}$$

$$\text{Subject to} \quad \sum_{j=1}^{n+1} Y_{ij} = 1 \qquad i = 1, m$$

$$\sum_{i=1}^{m+1} Y_{ij}=1 \qquad j=1,n$$

$$Y_{ij} \leq X_{ij} \qquad i=1, m+1; j=1, n+1$$

$$Y_{ij}=0 \text{ or } 1 \qquad i=1, m+1; j=1, n+1$$

which expresses the following concepts:

1. The objective is to minimize the sum of the chosen Quality Scores, including all non-correlation penalties.

2. Each track (target) must correlate with one and only one report (track) or be uncorrelated.

3. A track/target pair can correlate only if it has associated.

This optimization problem is a common type of transportation problem known as the assignment problem. The method of solution is well known, but unfortunately it involves an iterative procedure. In order to keep execution time within bounds, the exact solution will not be sought. Instead, the best first approximation to the solution will be used to select the correlation pairings. Simulations have shown that in virtually all cases the best first approximation and the final solution are identical. In fact, no case based on real data has yet been seen for which this hasn't been true.

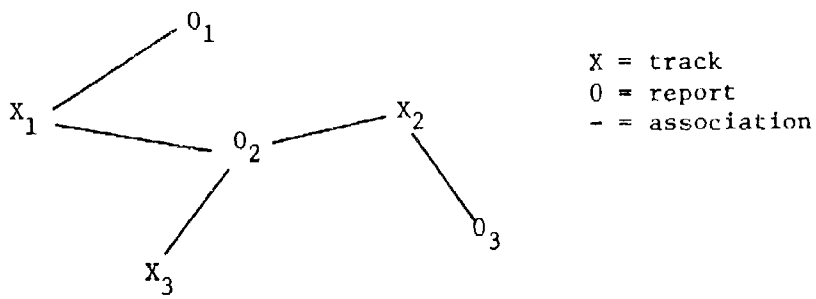
The first step in the resolution process is the formation of the lists of tracks and targets involved in the association system. This step is begun by placing any track on the first list and all of its associating targets on the second. Then all associating tracks of these targets are added to the first list, and all associating targets of the new tracks on the second list,

etc., until a closed set of tracks and targets has been found. Then, if any of these tracks or targets is ready to correlate, the resolution process begins; otherwise, all tracks and targets are flagged for carry over to the next sector (see Section 7.3). If correlation is to continue, a matrix of Quality Scores is constructed, with each track corresponding to a row and each report to a column. If a track and report do not associate with each other, the default score is entered into that element of the matrix. Figure 7-9 depicts such a matrix for a sample intertwined association case.

The heart of the resolution method is the order in which the tracks (or targets) are selected for correlation. Once a particular track (or target) is chosen, it correlates with its best remaining association partner. Then these two entities are eliminated from the group, and the next track (or target) is picked. The selection process utilizes targets if there are fewer reports than tracks, and tracks otherwise. By working on the minority entity, the possibility of correlating a fruit report or track is greatly reduced. This results because all minority entities are likely to be real, while the larger number of opposite entities is generally due to extraneous items. Thus, it is hoped that no fruit item will be correlated, as each selected minority member will choose a real partner. If the majority members were the selected entities, it is possible that a fruit entity would be selected before a real one, and thus it would form an incorrect correlating pair. This issue will be illustrated below in the example.

Assume for ease of discussion that tracks are the minority members. Then the track that is chosen next to correlate is the one that has the <u>most to lose</u> by not getting its first choice. To perform the selection, the difference in score between the lowest two Quality Scores in each remaining row is computed. The row with the largest such difference is the one selected. If a tie exists between two rows, the Deviation Scores for the entries in each row are employed. The track corresponding to the winning row is then correlated with the target corresponding to the lowest Quality Score in the row (Deviation Scores break ties). Finally, all the scores in the row and column of the selected pair are set to default, and the next selection is made. The process terminates when all rows have been chosen or when the winning correlation score is default. In the former case, all tracks have been correlated, while in the latter case, all remaining tracks must be left uncorrelated as all their associating reports have already been taken.

The resolution of a sample situation is illustrated by Figure 7-10. The track to target associations, the corresponding Quality Score matrix, and the initial row differences are all shown in part (a) of the figure. Since row 2 has the largest difference, track 2 is selected, and it correlates with target 3. The revised matrix for the next step is shown in part (b) of the figure. Rows 1 and 3 have equal differences, so Deviation Scores are required. When they are employed, row 1 is selected, and track 1 correlates with target 1. Finally, track 3 is last to be selected, and it correlates with target 4. It should be clear that this resolution in fact was the optimum one.

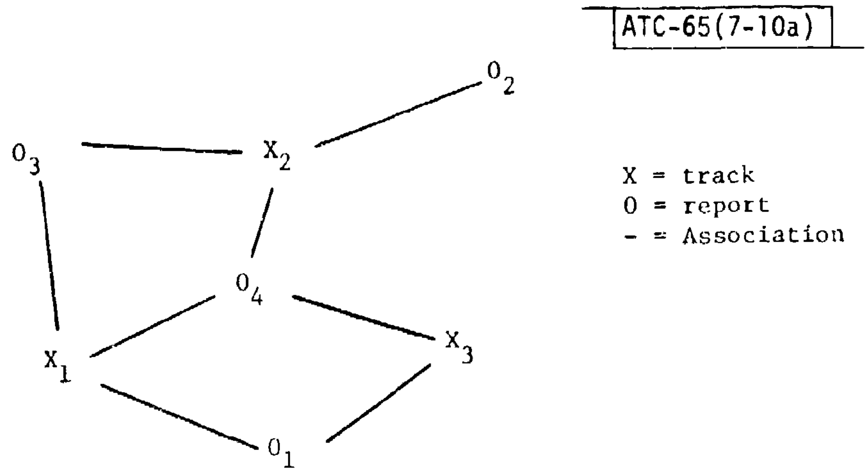


	Report 1	Report 2	Report 3
Track 1	30000000	10200000	50000000
Track 2	50000000	00420000	12300000
Track 3	50000000	20122000	50000000

Each entry is the octal Quality Score for the corresponding association.

Non-associating pairs, such as track 1 and report 3, are assigned the default score, 50000000.

Figure 7-9: Intertwined Association Matrix



Initial Quality Score Matrix:

	O_1	O_2	O_3	O_4	
X_1	00	50	10	20	Δ_1 = 10 - 00 = 10
X_2	50	21	00	20	Δ_2 = 20 - 00 = 20
X_3	20	50	10	00	Δ_3 = 10 - 00 = 10

(for simplicity, only 1st 2 octal digits of each quality score are shown; others are zero)

Δ_2 is largest, thus track 2 chooses first.

X_2 - O_3 score is smallest, thus track 2 correlates with report 3.

Figure 7-10a: Intertwined Example

Matrix after $X_2 - O_3$ correlation:

	O_1	O_2	O_3	O_4	
X_1	00	50	50	20	$\Delta_1 = 20 - 00 = 20$
X_2	50	50	50	50	$\Delta_2 = 50 - 50 = 00$
X_3	20	50	50	00	$\Delta_3 = 20 - 00 = 20$

Δ_1 and Δ_3 are tied, need Deviation Scores to decide who chooses first.

Assume Δ_1 is larger than Δ_3 after Deviation Scores are added to each matrix element.

Then track 1 chooses first, and selects report 1.

Matrix after $X_1 - O_1$ correlation:

	O_1	O_2	O_3	O_4	
X_1	50	50	50	50	$\Delta_1 = 50 - 50 = 00$
X_2	50	50	50	50	$\Delta_2 = 50 - 50 = 00$
X_3	50	50	50	00	$\Delta_3 = 50 - 00 = 50$

Track 3 chooses report 4

Figure 7-10b: Conclusion of Intertwined Example

For comparison, this example is redone in Figure 7-11 by allowing targets (columns) to be the selected entity. As is seen, in this case target 2, the extraneous report, correlates incorrectly with track 2. This happened because target 2 had only one associating track, and thus had the most to lose. In fact, fruit reports (or tracks) will often have only one association. By selecting minority entities, the problem of improper correlations due to fruit should be minimized.

After the set of correlations has been identified, each pairing is checked to determine whether or not it is ready to correlate according to the timing criteria of Section 7.3. If it is, the correlation is performed; otherwise, both the track and report are carried over to the next sector. Leftover tracks and reports are handled as above for the m on 1 and 1 on n cases (see Section 7.4).

The most common intertwined association situation involves two tracks and two reports. For this special case, the entire resolution algorithm reduces to the following comparison:

$$Q_{11} + Q_{22} \text{ vs. } Q_{12} + Q_{21}$$

If the first Quality Score sum is smaller, track 1 is correlated to target 1 and track 2 to target 2. If the second sum is smaller, the alternate pairing is chosen. Ties, as usual, are broken through Deviation Scores. If either selected Quality Score is the default value, that pairing is forbidden, and only one correlation will result.

Numerous other special intertwined situations could be resolved through short cuts. For example, a check could be made to see whether each track could be assigned its first choice report. If so, the correlations could be made directly. However, non-2 or 2 cases are so rare that the additional code to handle any other special case wouldn't be justified.

Initial Quality Score Matrix (refer to Figure 7-10 (a)):

	0_1	0_2	0_3	0_4
X_1	00	50	10	20
X_2	50	21	00	20
X_3	20	50	10	00
	$\Delta_1 = 20$	$\Delta_2 = 27$	$\Delta_3 = 10$	$\Delta_4 = 20$

Δ_2 is largest, thus report 2 chooses first and selects track 2

error!

ATC-65(7-11)

Figure 7-11: Redone Intertwined Example

8.0 TRACK INITIATION

ATCRBS tracks are automatically initiated when a pair of uncorrelated reports are found on successive scans that appear to have come from the same aircraft. These reports, to satisfy this criterion, must agree or potentially agree on both identity code and altitude. In addition, their physical separation must be sufficiently small that a real aircraft could have traversed the distance in one scan.

Not all uncorrelated reports enter into the track initiation process. Under user control, various categories of reports that are judged not likely to be due to real aircraft can be eliminated. The remaining uncorrelated reports are compared with those from the previous scan. If one or more matches are found, the report is used to start new tracks; otherwise, the report is added to the uncorrelated report buffer for comparison with subsequent scan reports.

If a current report is matched with more than one previous report, each potential new track is rated into one of four categories. Only those tracks in the highest category found will be initiated. If more than one track is created with the same current report, or more than one track created with the same previous scan report, this set of tracks will be linked together. Then, when the track corresponding to the real aircraft is identified (one report can only correspond to one aircraft), the other tracks are immediately dropped without having been declared in the system output.

8.1 Uncorrelated Target Buffer

Entries for all active uncorrelated reports are stored in the uncorrelated target buffer. Each entry contains the range, azimuth, identity code and code confidence, and altitude, altitude confidence, and altitude type fields of the original target report. The entries are linked according to the sector in which the report was created in order to provide an azimuth sorting capability. In addition, as explained below, this linking provides an easy method of determining which entries are no longer required. Figure 8-1 depicts the form of this buffer and its linking mechanism.

By the time a target is declared to be uncorrelated, it may have been in the system for several sectors. This occurs, as described in the previous chapter, when the correlation decision must be delayed. The worst case delay, controlled by a system parameter, can be as much as half a scan. Each new uncorrelated report attempts to locate uncorrelated reports from the previous scan that lie near its position. This search window will be centered at its position, and could have an azimuth extent as large as half a scan in each direction if it were very close to the sensor. Thus, the oldest required uncorrelated target will be two scans old, computed as follows:

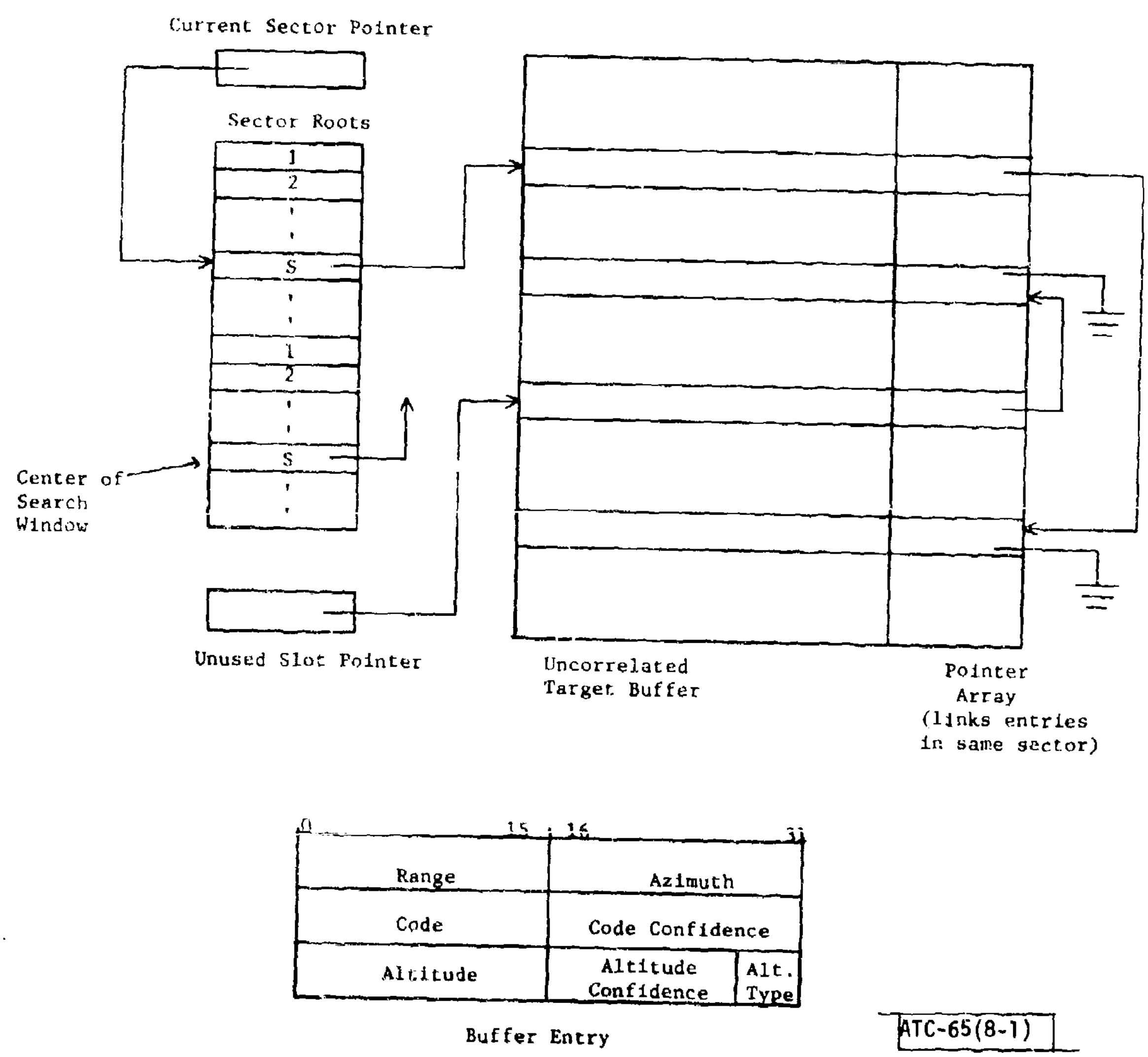


Figure 8-1: Uncorrelated Target Buffer

0.5 scan - delay period for current report

1.0 scan - search window center relative to current report position

$\frac{0.5 \text{ scan}}{2.0 \text{ scans}}$ - earliest edge of search window relative to the center

This fact accounts for the number of linked lists required in the uncorrelated target buffer: two per sector.

The linked list pointers are thus used in a circular manner. After the track initiation process is completed for the current sector, reports from this sector received two scans ago are no longer required. The pointer root for those reports is then free to be used for the current ones. Thus, each entry in the pointer root array always references reports in the same sector, making it very simple to determine the identity of the root for any given sector.

A separate linked list ties together all available slots in the buffer. When new reports are added to the buffer, the slots at the head of this list are utilized. The list is updated after every sector by adding to it the slots of all entries no longer required, namely those that are two scans old. This mechanism is needed because, unlike for the reply buffer of Chapter 4, entries in the same sector are not co-located in the buffer; correlation delays cause the set of uncorrelated targets for a sector to arrive piecemeal over a span of several sectors.

The final pointer associated with the buffer references the root linkage pointer for the current sector. This variable is required to indicate which of the two pointers for the sector is the current one. The other root pointer for the sector then serves as the center of the search region for non-delayed reports. The search center for delayed reports is offset back from this pointer by the number of sectors of delay. This search procedure is discussed further in Section 8.3.

8.2 Track Initiation Criteria

The track initiation process attempts to locate pairs of qualified uncorrelated reports with which to start new tracks. Uncorrelated reports that are judged to be due to fruit or system errors rather than to real aircraft are suppressed. This action not only prevents the formation of extraneous tracks but also significantly reduces the execution time of the process.

The types of reports that can be prevented from forming tracks, each under parameter control, are the following:

1. 1-hit reports (mode A or C)

2. 2-hit A/C reports

3. Leftover code swapping reports

4. Boresite reports

Even in fruit environments so low that 1-hit reports can profitably be used for tracking continuity during fades without overloading the system, an uncorrelated 1-hit report is more likely to be due to fruit than a real aircraft entering the system. Thus, such reports should probably be suppressed for best performance. On the other hand, in heavier fruit environments, where 1-hit reports are not created, reports formed by fruit replies will generally consist of two replies, one mode A and one mode C. This is because only coincidental position agreement is required for two such replies to correlate, while code agreement as well is required for replies of the same mode. In either case, if real reports are suppressed, the only system effect will be to delay slightly the formation of the track, as normal reports should be created on subsequent scans.

A report that was a candidate for code swapping, that is, one that lies very near another report in range and azimuth (see Section 4.4), is often caused by code declaration errors or fruit (see Section 6.4). Whether or not code swapping actually occurred, if such a report failed to correlate while its partner succeeded, the evidence is strong that the report is in fact extraneous. Thus, such reports should be suppressed.

Finally, boresite reports are often symptomatic of system errors, heavy garble, or sidelobe interference. Even if such reports corresponded to real aircraft, they could profitably be suppressed, as the tracks they initiated could have serious heading errors due to their uncertain azimuths. Unfortunately, one other cause of boresite reports exists in this implementation: an aircraft transponder that produces slightly wide pulses. If the pulses are just the right width, no monopulse samples will be taken on them by the reply processor.

Since this latter effect will persist for the life of the aircraft, its track would never be initiated if uncorrelated boresite reports were discarded. Thus, the modified rule to be employed is:

> permit two uncorrelated boresite reports to initiate a track, but reject any potential tracks consisting of one boresite and one monopulse report.

Current scan boresite reports that fail to form a track with previous scan ones are placed in the uncorrelated report buffer, but are not included in the output stream. Thus, most such reports are discarded eventually. Those boresite reports successfully initiating a track are of course reported out at once.

The first check made on a potential pair of track initiating reports is that their positional difference is sufficiently small to correspond to the motion of a real aircraft. Two box sizes as shown in Figure 8-2 are defined for this purpose, one corresponding to "normal" aircraft and one for unusual aircraft (military jets, SST's, etc.). A pair of reports is said to be in zone 1 if their differences satisfy the smaller limit and zone 3 if they satisfy the larger limit:

Zone 1: $\Delta\rho \leq \delta\rho_{small}$

and $\rho\Delta\theta \leq \delta\rho_{small}$

Zone 3: $\Delta\rho \leq \delta\rho_{large}$

and $\rho\Delta\theta \leq \delta\rho_{large}$

and not in zone 1

where ρ is the range of the current report. A potential pair satisfying neither test is rejected. Note that these tests are approximations to the circular test required and do not use ground range. Thus, they can fail for a high flying aircraft over the sensor. However, few if any tracks will be initiated in that region and at worst the track will be started one or two scans late.

Each successful pair is then checked for identity code and altitude agreement. This is done by computing ΔC and Δh for the pair by the same methods used for comparing targets against tracks for association in Section 6.2. Once these entities are known, the final zone of the pair is found from the geometric zone defined above as follows:

Zone stays the same if:

$\Delta C \leq \frac{1}{2}\Delta C_{max}$

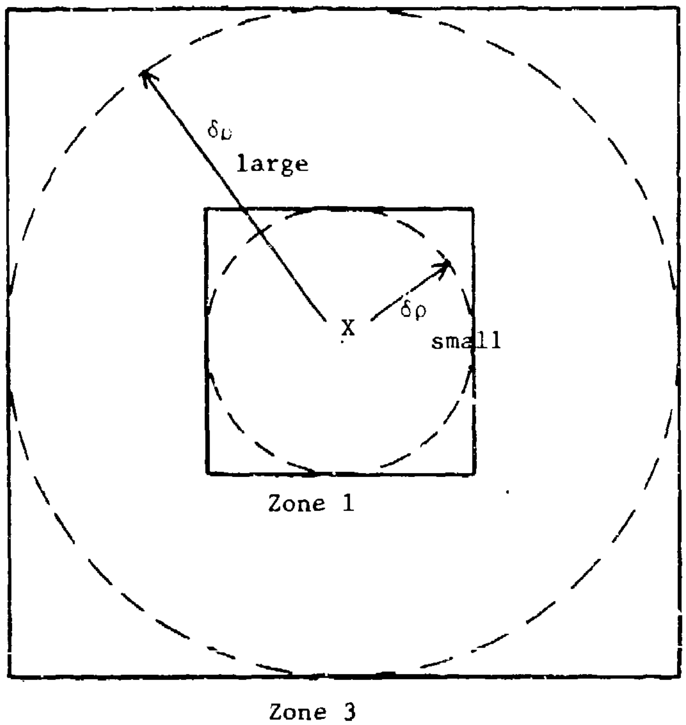
$\Delta h \leq \frac{1}{2}\Delta h_{max}$

Zone increases by 1 if:

$\Delta C \leq \Delta C_{max}$

$\Delta h \leq \Delta h_{max}$

and above conditions failed



X = report position

= locus of points that can be reached by aircraft flying at:
Zone 1: "normal" speeds
Zone 3: exceptional speeds

ATC-65 (8-2)

Figure 8-2: Track Initiation Boxes

If $\Delta C > \Delta C_{max}$ or $\Delta h > \Delta h_{max}$, or if the final zone is 4, the pair is rejected as a candidate for track initiation.

Thus, there are three categories of candidate track initiation pairs that are acceptable. Those target pairs in zone 1 fall within the normal geometric box relative to each other and agree on both identity code and altitude, those in zone 2 also fall within the small box but only potentially agree on either code or altitude, and those in zone 3 fall within the large geometric box and agree on both code and altitude. All other pairs of uncorrelated reports, one from the current scan and one from the previous scan, are rejected for track initiation.

8.3 Overall Track Inititation Algorithm

After target to track correlation is completed, each target report in the sector list is examined in order. If the report was correlated, it is passed over at this point and will be processed further during track update. If it is flagged to indicate that it is required for correlation in the subsequent sector (see Section 7.3), it is placed in the target list for the next sector and its output delayed accordingly. All other uncorrelated reports are examined to determine whether or not they are qualified to partake in track initiation. Those found unqualified are discarded as due to fruit or system error and are not output, while those passing the test are entered into the track initiation process. Whether or not these latter reports start a new track, they are output as uncorrelated. This is to prevent tracks from being declared to the outside world until a third, confirming, report is encountered.

When a qualified uncorrelated report is identified, the track initiation process, outlined in Figure 8-3, begins by determining which sectors of the previous scan must be examined in order to locate potential pairing reports. Denote the current sector by S_{curr}, and let θ_c and ρ_c be the azimuth and range respectively of the current report. Then the sector in which this report was created is given by

$$S_c = \frac{\theta_c}{\theta_{sect}} + 1$$

where θ_{sect} is the size of a sector and integer division is assumed. S_c will equal S_{curr} if the report was not delayed by the correlation process. Thus, if there are NS sectors in a scan, the center of the search region occurs NS+(S_{curr}-S_c) sectors prior to the present one. Since a pointer in the uncorrelated report buffer references the current sector linked list, the linked list for the search center is obtained by decrementing this value (in a circular fashion) the required amount. Finally, the number of linked lists on either side of the search center that must be processed is given by:

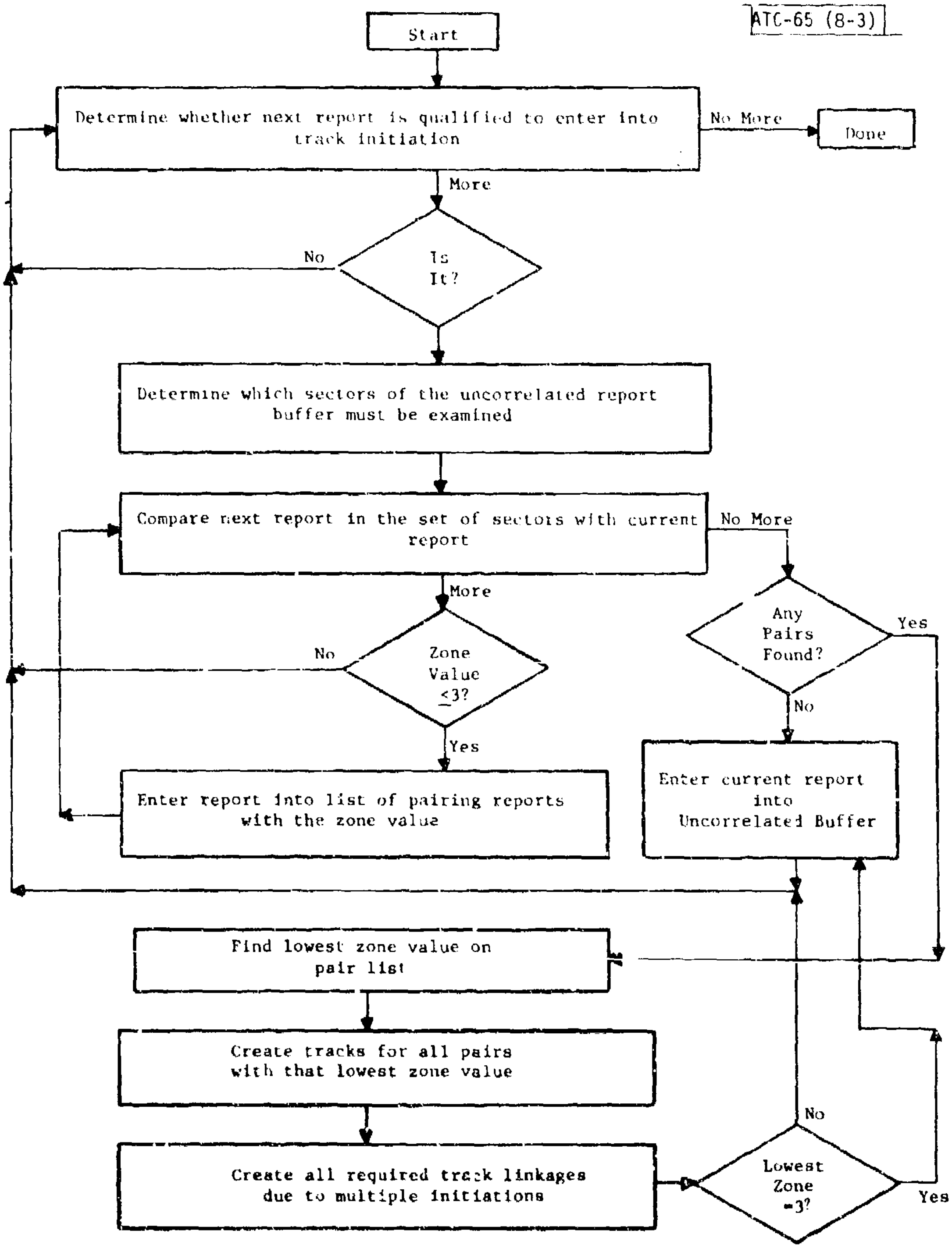


Figure 8-3: Track Initiation Process

$$\Delta S = \mathrm{Min}\left\{\frac{NS}{2}, \left(\tan^{-1}\frac{\delta\rho_{large}}{\rho_c - \delta\rho_{large}}\right)/\theta_{sect}\right\} + 1$$

as shown in Figure 8-4. Except when ρ_c is very small, the arctangent function can be approximated by its argument.

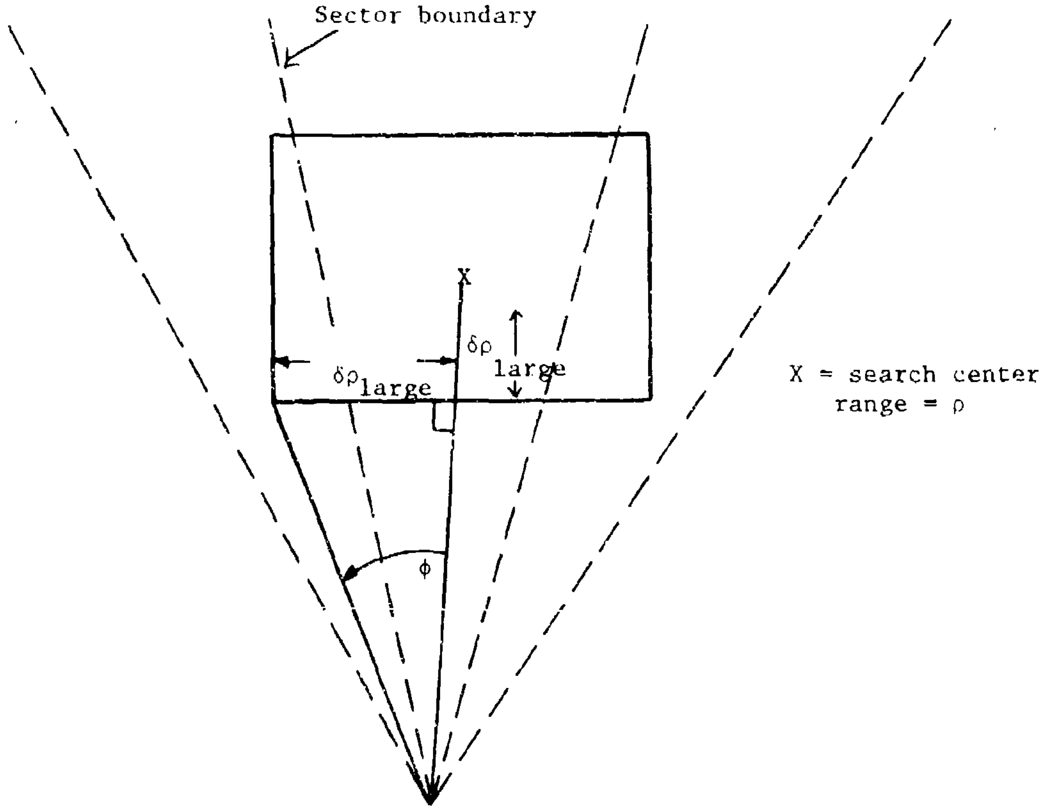
Each previous scan uncorrelated report existing on the linked lists of the sectors included within this search area is compared with the candidate current report. Using the procedure described in the last section, each such report is either discarded or entered into a list of pairing reports along with its zone value (1, 2, or 3). After all potential reports have been examined, the minimum zone value on the list is determined. All pairing reports that possess this value will be used to initiate tracks with the candidate report, while all pairing reports with higher values will be rejected.

If no pairing report was located during the search, the current report is entered into the uncorrelated report buffer and linked onto the proper sector list. It will then be available for pairing with uncorrelated reports received during the next scan. If the new report started one or more new tracks, but all these tracks were in zone 3 and therefore suspect, the report will also be entered into the buffer. This will permit the formation next scan of the correct track for the report if in fact the report were the first emanating from a new aircraft. If, however, the report is used to start one or more good tracks (zone 1 or 2), it is not entered into the buffer as it is very probable that one of these tracks corresponds to its aircraft.

The algorithm described above permits one current report to initiate more than one new track. Also, one uncorrelated report from the previous scan can be used by more than one current report to form tracks. Since any one report can only correspond to one aircraft, it is clear that in such cases extraneous tracks have been formed. Although the proper track of the set is not known at initiation time, it will become evident on a subsequent scan. This is because only the real one will be correlated on future scans (except for cases of coincident correlation of extraneous tracks and fruit reports). Thus, when one track of the set is correlated and the others coasted, these latter ones should be dropped at once to prevent erroneous future correlations.

In order to be able to identify all tracks in such a set, they must be linked together. The mechanism for creating these linkages is composed of the following rules:

1. If a current uncorrelated report initiates more than one track, by pairing with more than one previous scan report, all of these tracks are linked together. The current report is notified of this chain of tracks, but none of the previous reports are made aware of the track they helped to form.



$$\tan \phi = \frac{\delta\rho_{large}}{\rho - \delta\rho_{large}}$$

Maximum number of sectors required prior to center sector is thus:

$\Delta S = \frac{\phi}{\theta_{sect}}$ rounded up to next integer, which occurs if search center is just at sector's left border.

Figure 8-4: Track Initiation Search Extent

2. If a current uncorrelated report initiates only one track, and it is zone 1 or 2, the previous scan report is made aware of this track. If that report is already aware of other reports it has formed, the new track and those previous tracks are linked together.

3. If a current uncorrelated report initiates only one track, and it is in zone 3 (thereby implying that the current report will be available for additional track initiation next scan), only the current report is made aware of this track.

This set of rules guarantees that all tracks in a linked set have one report in common, and thus that only one can be real. Figure 8-5 illustrates several examples of the applications of these rules. Note that alternative groupings of tracks were possible in some of these cases. The only reasons for selecting the above rules over other possible sets were designer preference and implementation simplicity.

The field format for a track file entry is provided by Figure 8-6. The next chapter will discuss the use of the less obvious parameters. The figure also indicates how the parameters of the two target reports are used to initiate this file. The predicted position and velocity values for next scan will be developed during this scan's track update procedure, into which all newly initiated tracks are entered.

Finally, if the new track has a discrete 4096 identity code, the track must be entered into the discrete code array. Chapter 5 presented the method to be followed in such a case.

ATC-65 (8-5)

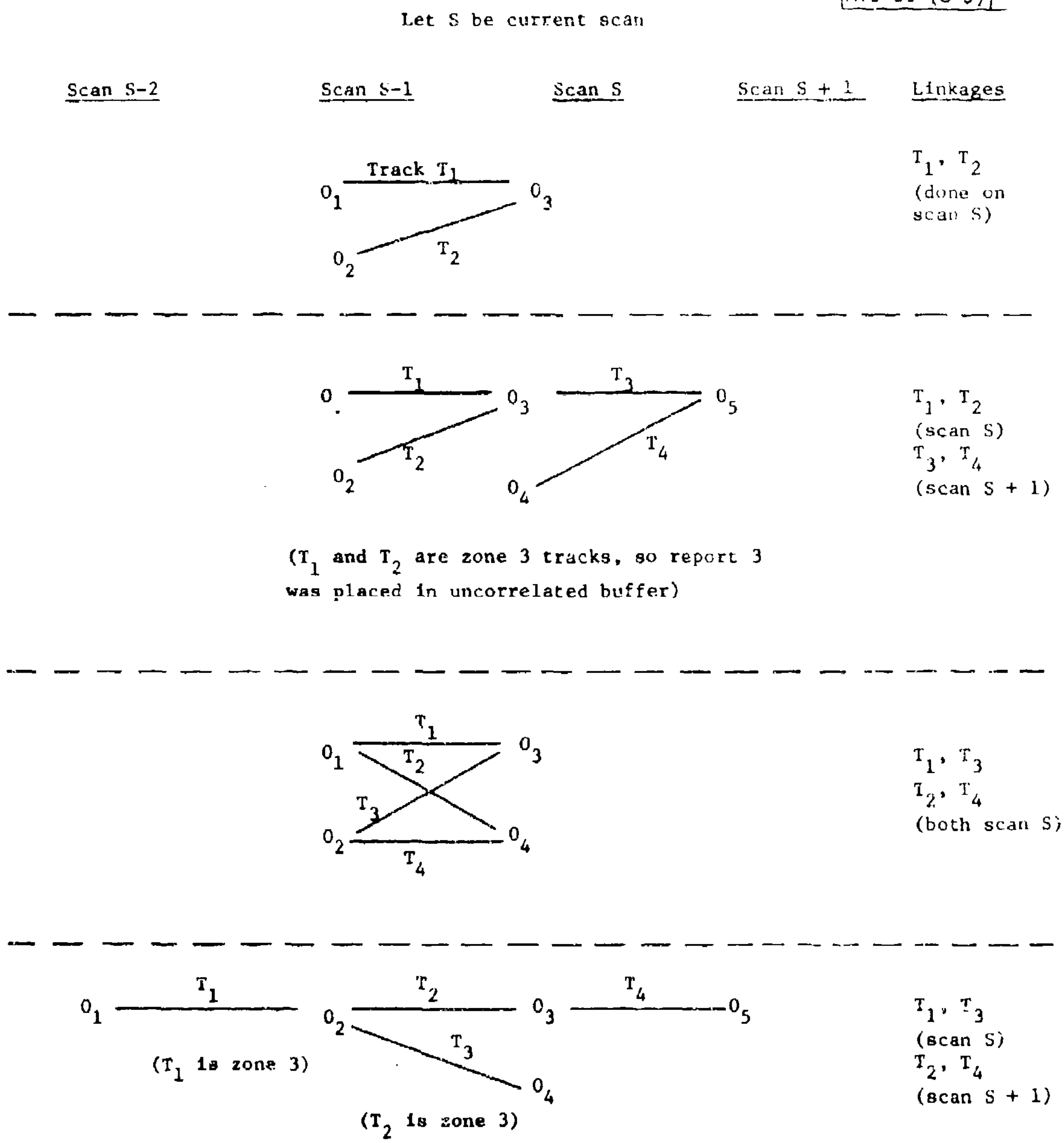


Figure 8-5: Track Linkage Examples

Track File Entry:

<table>
<tr><th>0 3</th><th>4 7</th><th>8 11</th><th>12 15</th><th>16 19</th><th>20 23</th><th>24 27</th><th>28 31</th></tr>
<tr><td colspan="4">Range</td><td colspan="4">Azimuth</td></tr>
<tr><td colspan="4">Official Code</td><td colspan="4">Official Confidence</td></tr>
<tr><td colspan="3">Altitude</td><td>Altitude Miss Count</td><td colspan="3">Altitude Confidence</td><td>Altitude Type</td></tr>
<tr><td colspan="4">Special Purpose Bits</td><td>Firmness</td><td>History Firmness</td><td colspan="2">Correlating Report #</td></tr>
<tr><td colspan="4">Range Rate</td><td colspan="4">Azimuth Rate</td></tr>
<tr><td colspan="4">Ground Range</td><td colspan="4">Ground Range Altitude</td></tr>
<tr><td colspan="3">Last Code</td><td>Code Miss Count</td><td colspan="3">Last Confidence</td><td>Turning State</td></tr>
<tr><td colspan="2">Track Life</td><td colspan="2">Cone Count</td><td colspan="4">Track Number</td></tr>
</table>

Initial Settings:

(R_1 = previous scan report, R_2 = current scan report)

Range and Azimuth: those of R_1
Official Code and Confidence: those of R_1
Altitude, Confidence, and Type: those of R_1
Altitude Miss Count: 0
Special Purpose Bits: according to track type (see page 2)
Firmness: 3
History Firmness: 1
Correlating Report #: number of R_2
Range and Azimuth Rates: 0
Ground Range Altitude: computed from Altitude
Ground Range: computed from Range and Ground Range Altitude
Last Code and Confidence: those of R_1
Code Miss Count: 0
Turning State: 0
Track Life: 1
Cone of Silence Count: 0

Figure 8-6: Track File Entry Format (1 of 2)

Special Bits

No. of Bits	Bit = 1 if	Reference Section
1	Test Track	-
1	Radar-only track	11.3
2	00 = track not dropped 01 = track dropped due to misses 10 = track dropped in cone of silence 11 = track dropped due to linkage	9.5
2	00 = track real 01 = track possibly false type I 10 = track possibly false type II 11 = track false	10.2
1	Track processed through correlation	7.3
1	Track coasted	9.5
1	Track has perfect association	6.2
1	Track updated by radar	11.2
2	00 = ρ,θ tracking used 01 = $\ddot{\rho},\ddot{\theta}$ tracking used 10 = X,Y tracking used	9.3-4
1	Track has discrete code	5.1
1	Track not yet mature	7.1
1	Linked track	8.3
1	Not active track	9.3-4

Figure 8-6: Track File Entry Format (2 of 2)

9.0 TRACK UPDATE

Each ATCRBS track has the information in its track file updated once per scan. If the track was correlated with a target report, the position and velocity predictions and the identity code and altitude values will all be modified according to the new data provided by that report. This report, in turn, will then be improved by using the many scan composite information available in the track file. Uncorrelated tracks, on the other hand, are merely coasted ahead one scan by using the velocity estimate contained in the track file.

In the normal situation, the track position and velocity predictions are made by interpolating ahead the last two target data points in ρ, θ coordinates. This type of tracker, known as a 2-point interpolator or an α=1, β=1 $\alpha\beta$ tracker, is sufficiently accurate for the short range predictions required for target to track correlations. Conflict detection or other long range estimation would of course require a more sophisticated tracking algorithm. A very rudimentary form of turn detection is added to this tracker to prevent fatal track deviations when potentially spurious data points are encountered.

When the track is near the sensor, however, the curvature of a ρ, θ coordinate system is too severe to be ignored. Thus, second order ρ, θ tracking (using accelerations) is employed at short ranges, and x, y tracking is used in the region surrounding the sensor. Since the latter type of tracking requires time-consuming coordinate conversion, it is only used where all forms of ρ, θ tracking are inadequate.

After a track is predicted ahead to the next scan, the sectors in which it will attempt to correlate are computed. The track is then placed on the linked list for the first such sector so that it will be activated at the proper time. The track will continue to move from sector to sector until it either correlates or arrives at the end of its search.

Figure 9-1 presents in flowchart form the series of operations that are performed on each track on the current sector's linked list. The remaining sections will present the detailed descriptions of each operation.

9.1 Track Code Update

An ATCRBS track file (refer to Figure 8-6) contains two identity code entries along with their corresponding confidence words: one that represents the official code of the track and the other that consists of the code of the last correlating target report. In general, these codes will agree with each other; disagreement occurs when an incorrect correlation is made or when an aircraft identity code is ordered changed by an air traffic controller. When these codes differ, a counter in the track file indicates how many successive correlations have produced codes that, although different from the official code, are self-consistent. When this counter reaches a parametric value, the new code replaces the previous official code in the file.

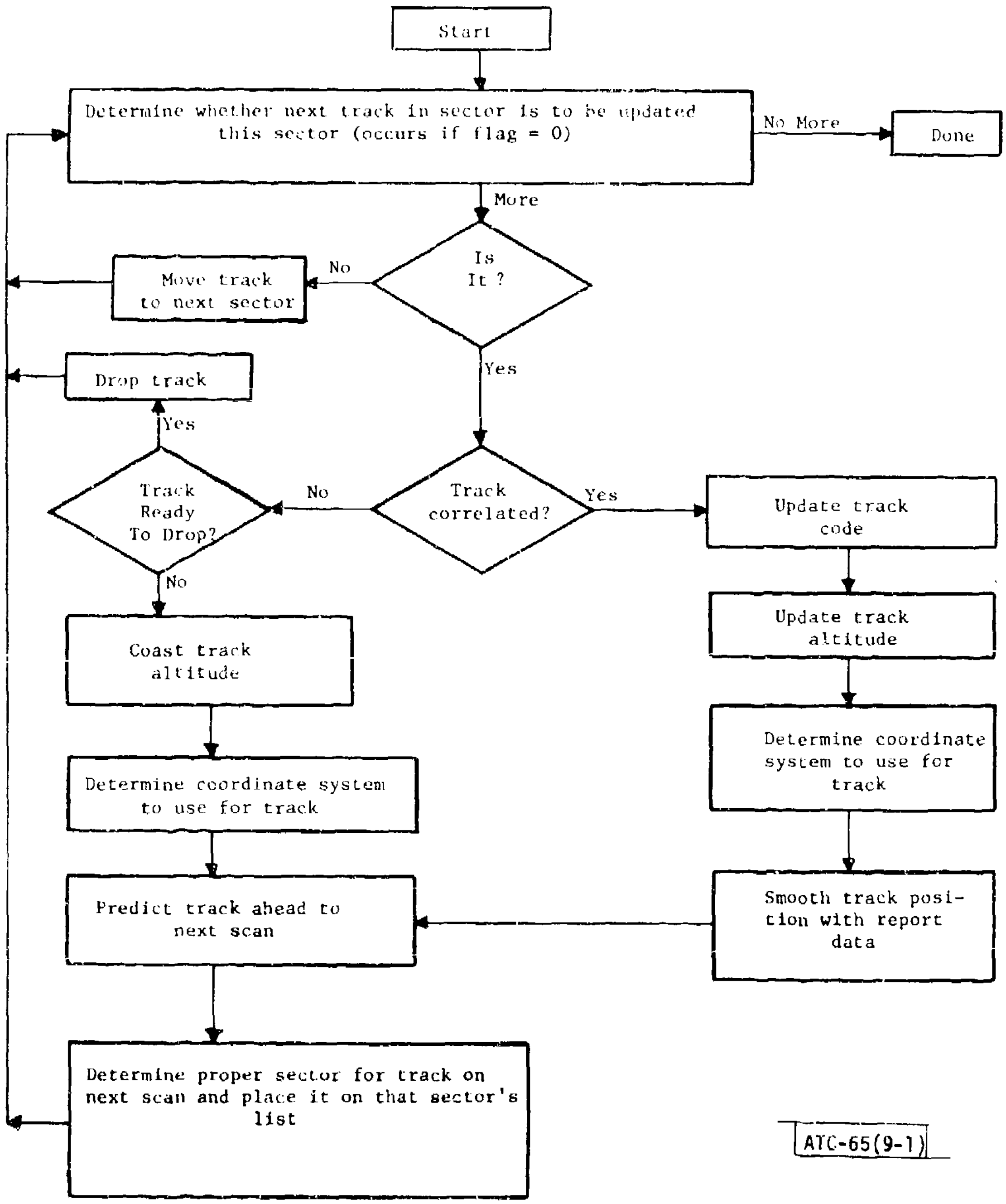


Figure 9-1: Track Update Overview

The first action in the code update process is to check whether the code of the correlating report agrees with the official track code. This determination has already been made during the association process, and the answer is contained in the score of the target/track association. As defined in Section 6.2, the zone-code value of the association is found as:

$$\text{zone-code} = \frac{\text{association score}}{\Delta h_{max} + 1}$$

using integer division. The report and track codes agree, or potentially agree, if the zone-code result is 1, 2, or 5. If any of these values is obtained, the official code is updated, the target report code and confidence are placed in the track file as the last correlated code, and the code counter is set to zero.

The code update formula creates a high confidence bit if either the track or target code was high confidence in that position, except that a low confidence '1' is created if both were high confidence and they disagreed (potential agreement implies less than a parametric number of such instances). Hopefully, the track code will become totally high confidence through this procedure even if the aircraft is continually garbled. The equations used are:

$$C \leftarrow C_{trk} \cdot C_{tgt} + C_{trk} \cdot \overline{F}_{trk} + C_{tgt} \cdot \overline{F}_{tgt}$$

$$F \leftarrow F_{trk}F_{tgt} + C_{trk}\overline{F}_{trk}\overline{C}_{tgt}\overline{F}_{tgt} + \overline{C}_{trk}\overline{F}_{trk}C_{tgt}\overline{F}_{tgt}$$

where C and F are the new official code and code confidence values for the track. These equations are the same as used for mode C update in reply correlation (see Figure 4-11b). After these code and confidence values are determined, they are written into the target report so that each report will contain the best estimate of the true aircraft identity code.

If the report code disagrees with that of the track, as indicated by a zone-code value of 3 or 4, the report code is compared with the last reported code entry in the track file. This comparison is identical to the code comparison calculation for target to track association. To review, the following syndrome sequence is computed:

$$S = \overline{(\overline{C_{last} \oplus C_{tgt}}) \vee F_{last} \vee F_{tgt}}$$

If $||S|| \leq P$, that is, if fewer than P '1's are in the syndrome, agreement is said to exist. In such a case, the last reported code and confidence fields

of the track file are updated by the same equations presented just above, and the code counter is incremented. If the counter value becomes equal to a parametric value, this code and confidence pair is used to replace the official track code and confidence fields of the track file, and a code change is said to have occurred for the track.

Finally, if the target code agrees with neither of the codes contained in the track file, the target code and code confidence words replace the last reported code and confidence entries in the track file, and the code counter is set to one. In this situation, or in the previous one of target agreement with the last code, the code in transition bit is set in the target report and its code is left unchanged.

If, through this code update process, the official code of the track file is altered in any way, either by being modified, improved, or replaced, it is possible that the discrete track file discussed in Section 5.1 must be modified. Two types of changes are possible. If the track had a discrete code prior to the alteration, its entry in the discrete file must be eliminated. Or, if the new code is discrete, an entry must be created for the track. If the track's code changed from one discrete code to another, then both of these actions are required. Section 5.1 explains the mechanism to be followed in each case.

9.2 Track Altitude Update

Each ATCRBS track has two altitude entries associated with it. The first entry, consisting of an altitude word, a confidence word, and an altitude type, provides the best guess of the current aircraft altitude value and is employed by the target to track association process. The altitude word is kept in flight levels (100's of feet) if all bits are declared with high confidence, but is left in unconverted Gray code form if any uncertainty exists. The second altitude entry provides the last known altitude level of the aircraft, in range units, and is used to compute ground range whenever necessary.

An aircraft, depending upon the sophistication of its transponder, can respond in three different manners to a mode C altitude interrogation:

1. No response of any kind

2. Brackets only, no code bits

3. Encoded Gray code altitude level

In the first case, the current altitude is set to all bits low confidence and the ground range altitude is set to the default value, which is a parameter nominally set at half a mile. However, the ground range altitude is

never permitted to be greater than half the slant range prediction of the track. In the second case, when no uncertainty exists, the current altitude is maintained in a special Gray code form, all high confidence zeros. The ground range altitude in this case is again set to the larger of the default value or half the slant range. Finally, when a true altitude response is provided by the aircraft, the current altitude estimate is set as described in the next paragraphs, while the ground range altitude is kept at the last altitude level known with certainty. If no all high confidence altitude has yet been received, the default value is utilized instead.

The track file also contains an altitude miss counter that is similar in function to the code counter. This counter records the number of successive scans for which no correlating target report has been received that confirmed the current altitude. Thus, this counter is changed whenever the track coasts (no correlating report found) as well as when the correlating report has an unknown or disagreeing altitude value.

The counter starts at zero and is incremented for each non-confirming scan until it reaches a parametric value. At that time, the altitude confidence field is set to all bits low confidence, indicating that the track altitude is no longer sufficiently current to be used with certainty in target to track association. This confidence field setting will permit any report to pass the altitude test, although those which agree with the altitude value will be scored much better. If additional non-confirming scans occur after this time, the counter decrements one unit for each scan until it reaches zero again. Should this event occur, the altitude and confidence entries of the current correlating report are placed into the track file and the entire cycle begins again.

The details of the various classes of altitude information that a track file can contain are presented in the Appendix. The update rules discussed above are also described there in greater detail.

9.3 Normal Position and Velocity Update

ATCRBS reports are expressed in a ρ, θ coordinate system. Target to track correlation is performed using ρ and θ values. Thus, the system would perform much more efficiently if tracking were also performed in ρ, θ, eliminating many otherwise useless coordinate conversions to and from x,y. The problem with this approach, of course, is that a ρ, θ coordinate system is not rectilinear. Thus, an aircraft flying in a straight line will not maintain constant ρ and θ velocities, which precludes making long-term track projections in terms of simple time-velocity products.

Tracking at the ATCRBS sensor, fortunately, is only used to permit proper target-to-track correlation. Thus, only short-term tracking accuracy, generally one scan into the future, is required. For such intervals of time,

and for aircraft not near the sensor, the assumption of constant ρ and θ velocities is quite good. Figure 9-2 indicates the magnitude of the one-scan ρ, θ prediction errors as a function of range for the worst case situation, namely a very fast aircraft (500 knots) flying tangentially to the sensor. As can be seen, the errors in ρ, θ tracking remain negligible for aircraft as close as five miles from the sensor for a 4-second scan.

Track update consists of two separate functions: smoothing and projection. Smoothing attempts to correct the present position and velocity estimates of the aircraft by blending together the track's predictions with the new target report data point. The most common method of smoothing, known as $\alpha\beta$, utilizes the following equations:

$$\rho_{smooth} = \rho_{pred} + \alpha(\rho_{tgt} - \rho_{pred})$$

$$\theta_{smooth} = \theta_{pred} + \alpha(\theta_{tgt} - \theta_{pred})$$

$$\dot{\rho}_{smooth} = \dot{\rho}_{pred} + \frac{\beta}{f}(\rho_{tgt} - \rho_{pred})$$

$$\dot{\theta}_{smooth} = \dot{\theta}_{pred} + \frac{\beta}{f}(\theta_{tgt} - \theta_{pred})$$

where the velocities are per scan quantities and f is the track firmness (number of scans since last data point). That is, a fraction α of the position error and β of the velocity error are employed for smoothing.

Larger values of α and β permit the track to follow aircraft turns more accurately and quickly, while smaller values eliminate erratic track behavior due to random noise for straight flying aircraft. The types of aircraft trajectories expected, the quality of the data, and the penalties incurred by tracking errors all contribute to the decision of what values to employ. In addition, the settings of α and β are often varied during the life of a particular track as a function of the coasts and maneuvers of the aircraft under track.

The present ATCRBS implementation has both α and β set to unity, thereby producing a tracker known as a two-point interpolator. This name is indicative of that fact that these values of α and β result in the data point being used as the smoothed position, and thus the track projection is based solely on the last two data points. This method of tracking was selected for two reasons: the monopulse capability of DABS is felt to provide high quality report position data, and immediate sensitivity to turns is desired. Ongoing analysis will be used to decide whether or not real world data quality is sufficiently accurate to justify these assumptions.

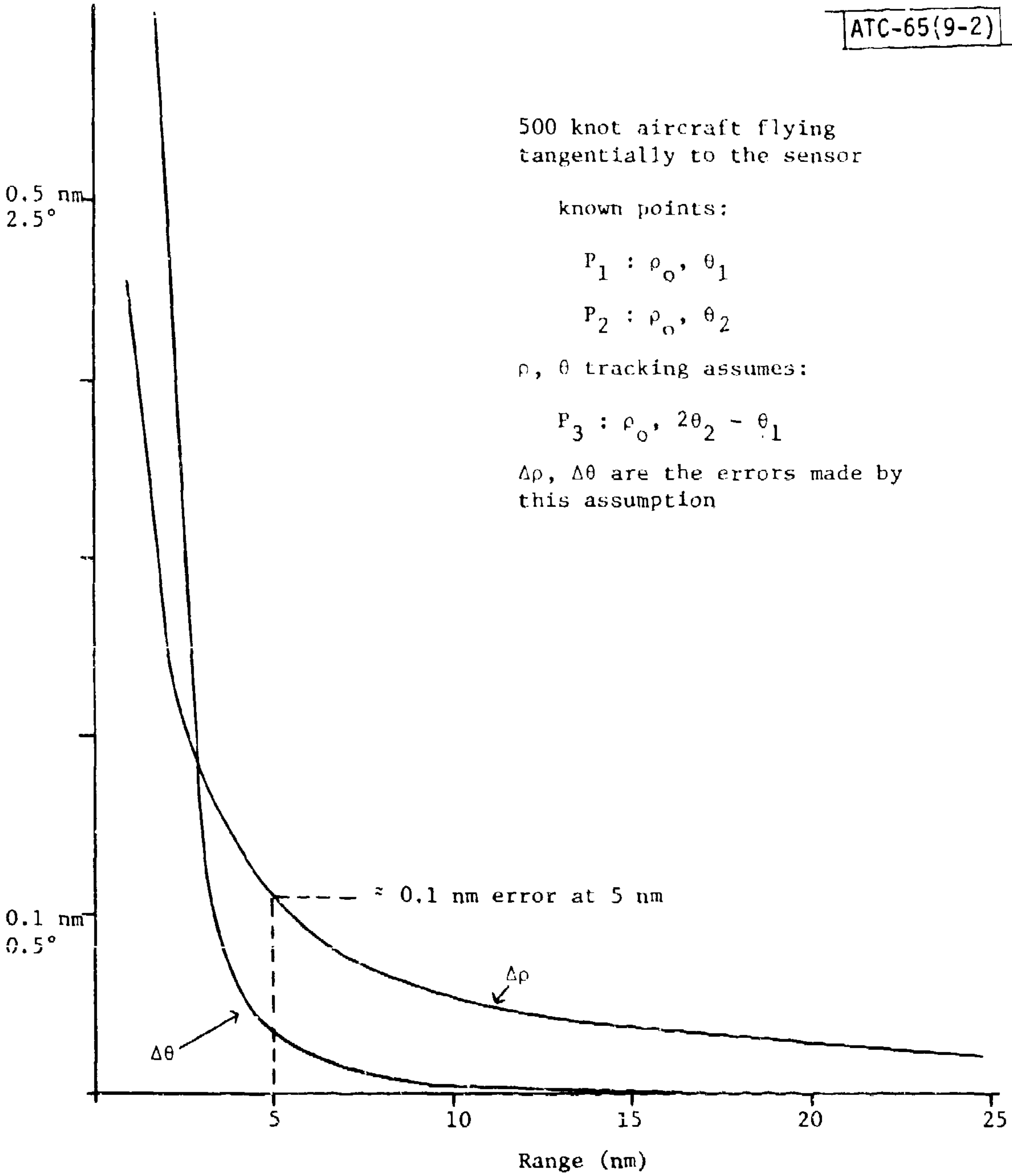


Figure 9-2: Worst Case ρ, θ Prediction Errors

There are two situations in which this simple smoothing rule is modified. The first occurs when the correlating target report does not contain a monopulse azimuth, that is, when its azimuth has been determined through boresight beamsplitting. The errors inherent in such an azimuth are too great to permit putting complete faith in its value. In this case, the azimuth of the report is modified as follows:

$$\theta'_{tgt} = \frac{\theta_{tgt} + \theta_{pred}}{2}$$

before smoothing is performed. This action is equivalent to using a setting of 1/2 for α and β, giving equal weight to the data and the prediction.

The second instance in which the data is not totally trusted, illustrated in Figure 9-3, occurs when a track that has received several successive good correlations from a straight-flying aircraft suddenly correlates with a target report far from its predicted position. Such a condition could indicate an erroneous correlation. In that event, full smoothing could cause the track to deviate sufficiently far from the true aircraft trajectory to result in its being subsequently dropped.

To prevent such a catastrophic occurrence, smoothing beyond the track's zone 1 association box (refer to Section 6.2) is not permitted for well-behaved tracks. Tracks subject to this rule are defined as follows:

1. The track has correlated on both of the previous two scans (call these scans n-2 and n-1).

2. The last correlating target report (on scan n-1) fell within the box 1 association region of the track.

3. The current correlating report, on scan n, falls outside of the box 1 region in either ρ or θ (or both).

When such a track situation is encountered, the following actions, depicted in Figure 9-4, are taken:

1. The track is smoothed in the offending coordinate(s) only to the limit of the box 1 zone.

2. An entry is made in the turning state field of the track file (see Figure 8-6) of the direction, positive or negative, of the target deviation in this coordinate(s).

Then, should the next correlating target report, on scan n+1, again fall outside of the zone 1 association box in the same direction as that on scan n, full smoothing is utilized on that scan. Furthermore, full smoothing is

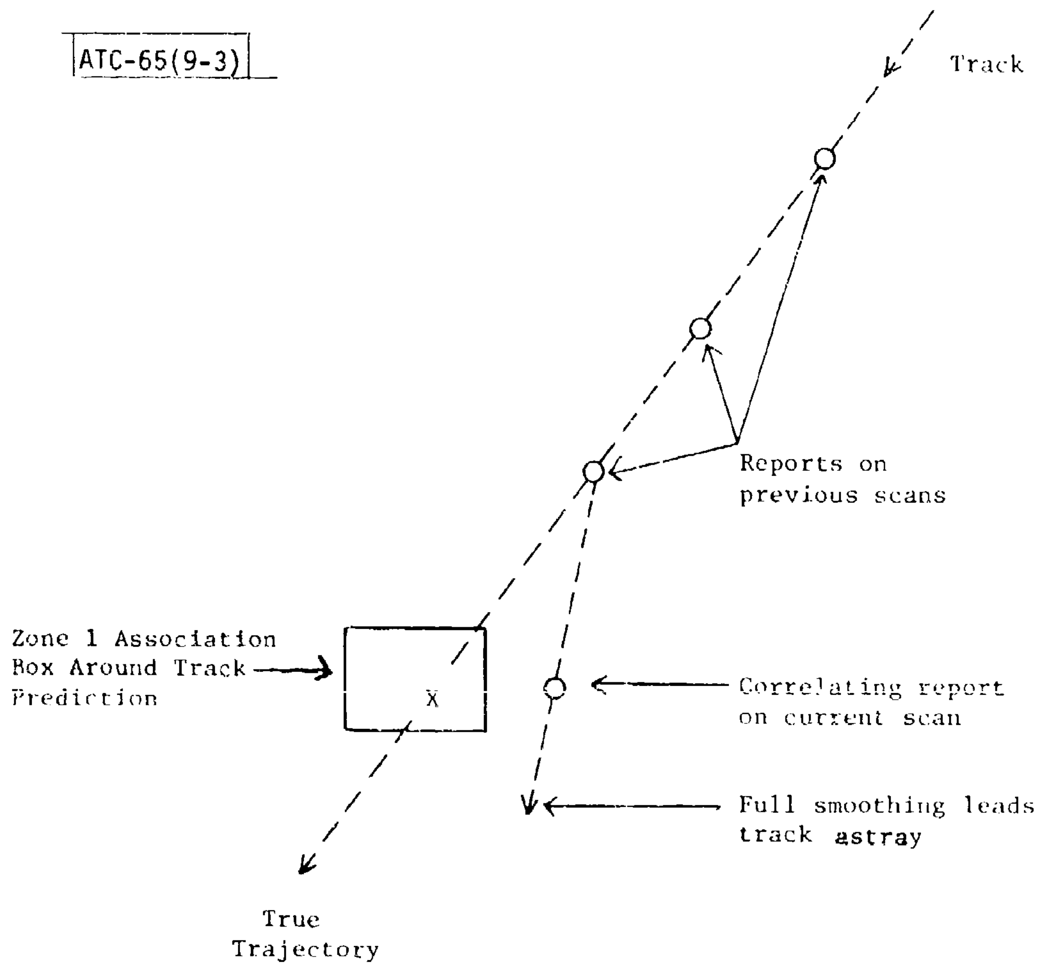


Figure 9-3: Suspect Smoothing Situation

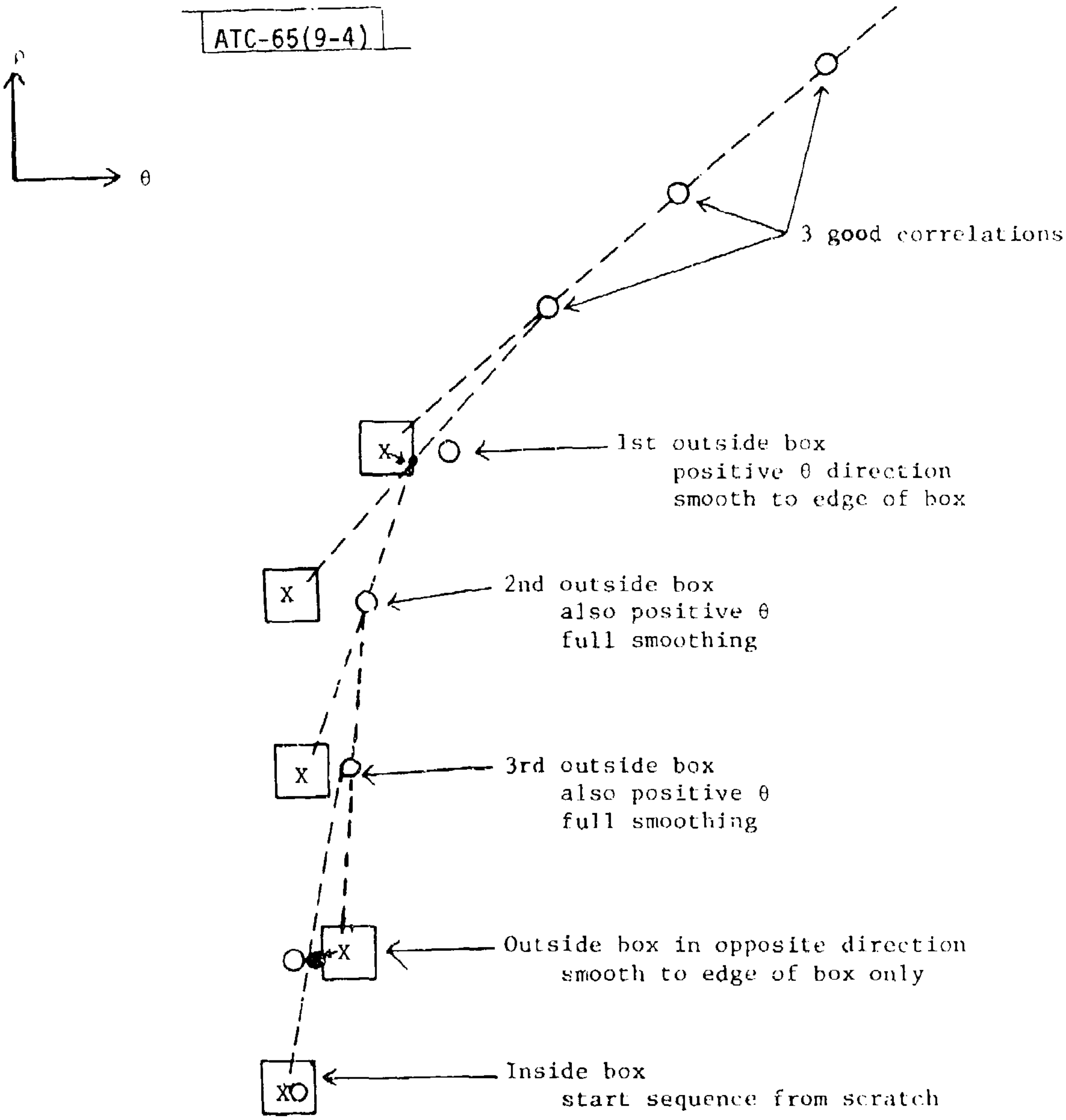


Figure 9-4: Turn Detection Smoothing Example

maintained for the duration of the aircraft turn, that is, as long as the reports fall outside of the box in that same direction. Once a report again falls within the track's box 1, the smoothing rule is reinitialized. Should the report on scan n+1 fall outside of the box in the opposite direction from the one on scan n, however, actions 1 and 2 above are again taken, and the new direction of deviation is recorded. To summarize, smoothing of a continuously correlating track beyond the boundaries of its zone 1 association box is only permitted when the previous scan's report fell outside of the box in the same direction as the current one.

This process thus implements a very crude turn detection mechanism. The first report in a turn is treated with suspicion, but once the turn is confirmed, the data points are followed fully. This mechanism hopefully will prevent erroneous track deviations at the cost of only a one scan delay in following aircraft turns. In addition, this algorithm will provide a degree of smoothing for tracks in diffraction situations. In such cases, data points tend to oscillate in azimuth. Since successive points then fall outside the association box in opposite directions, no data point is accepted at face value.

The second function of track update is the projection of the track's smoothed position to the expected location of the next target report. This operation is quite straightforward once the time until the reception of that report is known. For aircraft not near the sensor, such as those for which ρ, θ tracking is being utilized, this interval is almost exactly the time of one antenna revolution independent of the aircraft's tangential velocity. Thus, the new track predicted position is given simply as:

$$\rho_{pred} = \rho_{smooth} + \dot{\rho}_{smooth}$$

$$\theta_{pred} = \theta_{smooth} + \dot{\theta}_{smooth}$$

The final track file fields that require updating are the firmness f, history firmness g, and the track life. The first two quantities represent the number of scans since the last correlation and the number of scans between the last two correlations respectively. Thus, when a correlation has just occurred, as assumed in this section, the new value of f is 1. In the usual case, the new value for g is simply the previous value of f. However, if the track has just completed a coast through the sensor cone of silence, the new value of g is given by the number of such coasts added to the previous value of f. Section 9.5 discusses the cone of silence issue in detail. The track life field, which counts the number of reports in the track history, is simply incremented.

9.4 Short-range Position and Velocity Update

When an aircraft flies near the sensor, the errors inherent in simple ρ, θ tracking become sufficiently large that target to track correlation could no longer be supported. Thus, an improved method of tracking is required. Two alternative methods are possible: second (or higher) order ρ, θ tracking and coordinate converted x, y tracking. Both of these methods are utilized in the DABS system.

By introducing the acceleration terms $\ddot{\rho}$ and $\ddot{\theta}$ into the projection equations, much of the error inherent in simple ρ, θ tracking can be corrected. The new equations then become:

$$\rho_{pred} = \rho_{smooth} + \dot{\rho}_{smooth} \times \tau + \ddot{\rho}_{smooth} \times \frac{\tau^2}{2}$$

$$\theta_{pred} = \theta_{smooth} + \dot{\theta}_{smooth} \times \tau + \ddot{\theta}_{smooth} \times \frac{\tau^2}{2}$$

where τ is the time until the next target report is expected. The calculation of this interval, which can no longer be assumed to be equal to exactly one scan, is described below. Figure 9-5 presents the worst case tracking errors that occur with these second order ρ, θ equations. From this figure it is seen that this type of tracking, for a 4-second scan, can be employed between two miles and the five mile cutoff of the simple ρ, θ tracking.

The smoothing algorithm for improved ρ, θ tracking is identical to that presented above for simple ρ, θ tracking. In particular, both the boresight and erratic data point special cases are treated in the same manner, and α and β are both set equal to unity. Once the smoothed values of ρ, θ, $\dot{\rho}$ and $\dot{\theta}$ are determined, the values of the acceleration terms are computed from them as follows:

$$\ddot{\rho} = \rho_{smooth} * \dot{\theta}^2_{smooth}$$

$$\ddot{\theta} = -2\dot{\rho}_{smooth} * \dot{\theta}_{smooth}/\rho_{smooth}$$

Finally, the projection of the track to the next scan is accomplished by applying the equations specified above.

When a track is between two and five miles from the sensor in ground range, its tangential velocity can no longer be ignored. That is, its time between updates can be sufficiently different from the scan period to affect the prediction accuracy if $\tau=1$ were assumed. However, it is probably true that the track's tangential velocity will be nearly constant between updates. Thus, as shown in Figure 9-6, the correct value to employ for τ is given approximately by:

$$\tau = \frac{1}{1-\dot{\theta}/2\pi} \qquad \dot{\theta} \text{ in radians/scan}$$

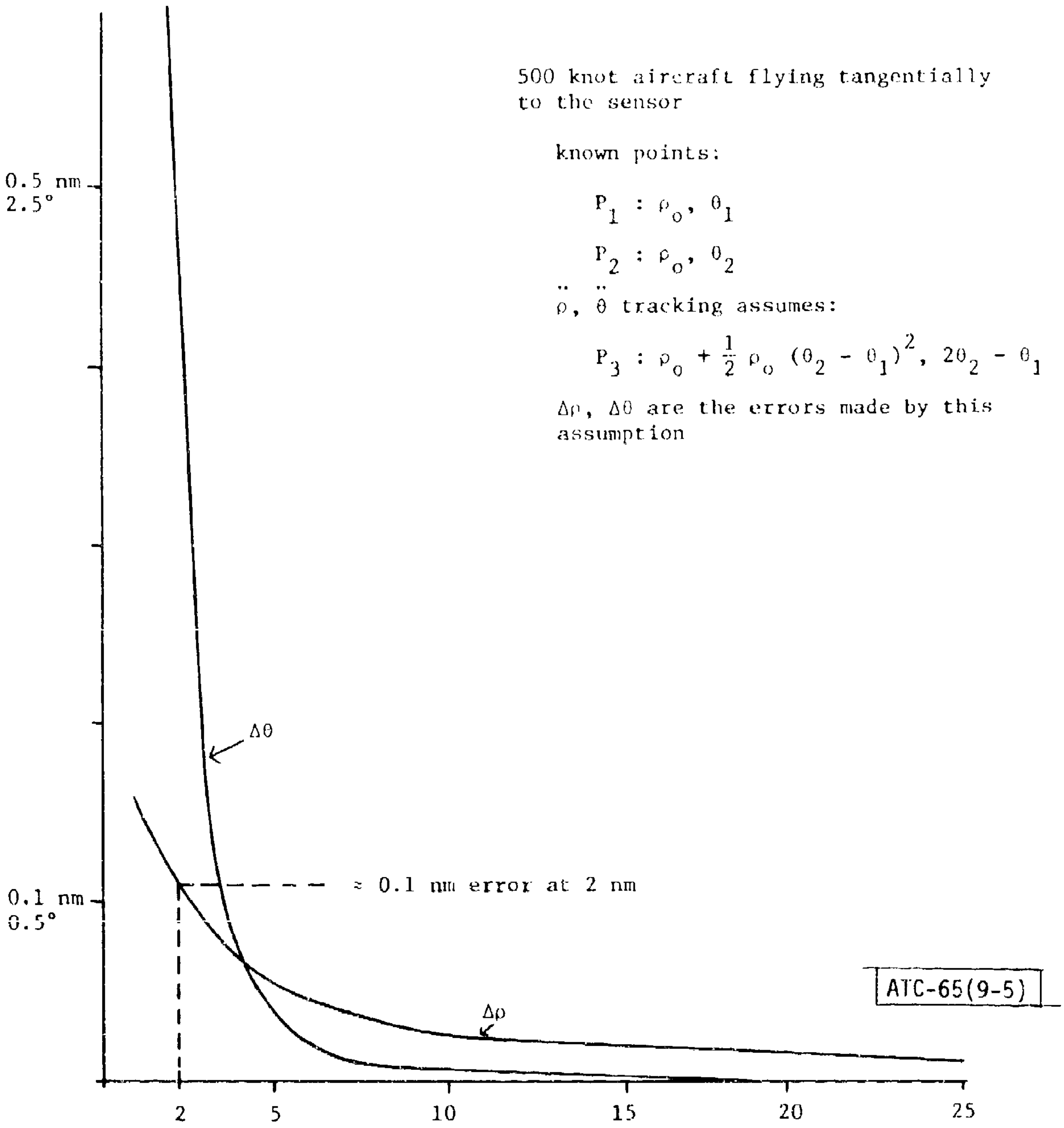
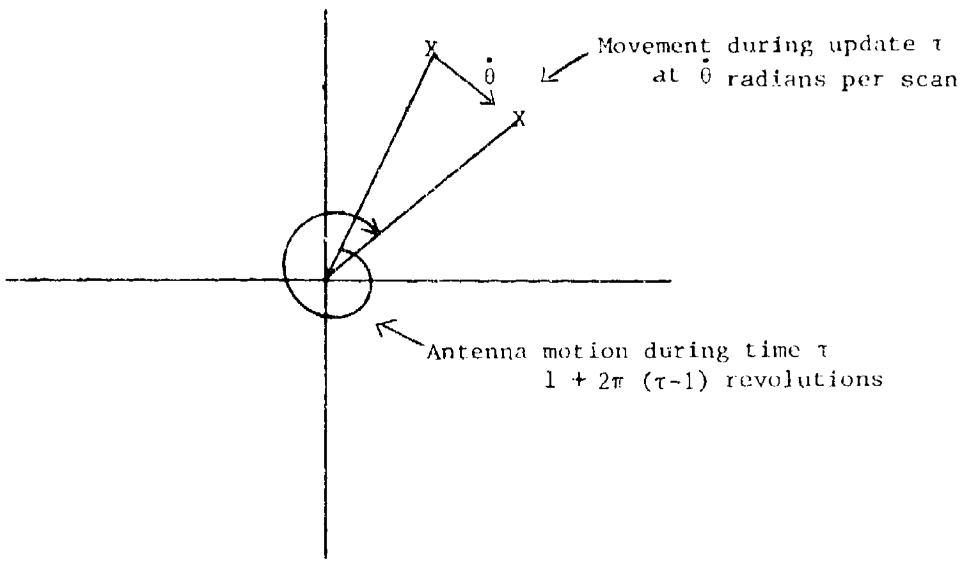


Figure 9-5: Worst Case $\ddot{\rho}$, $\ddot{\theta}$ Prediction Errors



Antenna catches aircraft:

$$2\pi\ (\tau-1) = \dot{\theta}\tau$$

$$\tau = \frac{1}{1 - \dot{\theta}/2\pi}$$

Figure 9-6: τ for Constant Velocity Aircraft

For aircraft very close to the sensor, no form of ρ, θ tracking can produce sufficiently accurate performance. Thus, even though it involves time-consuming coordinate conversions, x, y tracking must be employed for all such aircraft. For these tracks, the predicted position is maintained in both ρ, θ and x, y coordinates. Although not required, this extra storage eliminates the need for coordinate conversion of track data. Also, since $\dot{x}$ and $\dot{y}$ are the critical velocities for this mode of tracking, they are stored in the track file instead of $\dot{\rho}$ and $\dot{\theta}$. These latter velocities can be calculated whenever required as:

$$\dot{\rho} = \frac{x\dot{x} + y\dot{y}}{\rho}$$

$$\dot{\theta} = \frac{y\dot{x} - x\dot{y}}{\rho_{gnd}^2} \qquad \rho_{gnd} = \sqrt{\rho^2 - h^2}$$

The initial conversions from ρ, θ to x, y coordinates are given by:

$$x = \rho_g \sin\theta$$

$$y = \rho_g \cos\theta$$

$$\dot{x} = \frac{x\rho\dot{\rho}}{\rho_{gnd}^2} + y\dot{\theta}$$

$$\dot{y} = \frac{y\rho\dot{\rho}}{\rho_{gnd}^2} - x\dot{\theta}$$

The first action of the x, y track update process is the conversion of the target report coordinates:

$$x_{tgt} = \sqrt{\rho_{tgt}^2 - h^2} * \sin\theta_{tgt}$$

$$y_{tgt} = \sqrt{\rho_{tgt}^2 - h^2} * \cos\theta_{tgt}$$

where h is the internal track altitude (or the default value if its altitude is unknown). Track smoothing is then carried out in the same manner as for ρ, θ tracking, namely:

$$x_{smooth} = x_{pred} + \alpha(x_{tgt} - x_{pred})$$

$$y_{smooth} = y_{pred} + \alpha(y_{tgt} - y_{pred})$$

$$\dot{x}_{smooth} = x_{pred} + \frac{\beta}{f}(x_{tgt} - x_{pred})$$

$$\dot{y}_{smooth} = \dot{y}_{pred} + \frac{\beta}{f}(y_{tgt} - y_{pred})$$

Once again, the values of $\alpha=1$ and $\beta=1$ have been assumed at this point in the design validation process.

The two special cases of smoothing discussed above also apply for x, y tracking, although suitable modifications are required. In particular, if the target report has only a boresight azimuth, this value is smoothed prior to conversion in the following manner:

$$\theta'_{tgt} = \frac{\theta_{tgt} + \theta_{pred}}{2}$$

Then the regular x, y smoothing formulas are applied. The special smoothing that occurs when a suspect deviating report is found is treated just like for ρ, θ tracking. That is, the track is smoothed only to the limit of its zone 1 association box. For x, y tracking, this box is assumed to be of size ρ^1 in both the x and y coordinates, where ρ^1 is the ρ extent of the first ρ, θ association zone.

After smoothing is completed, the track is projected ahead to the next expected update position in the following manner:

$$x_{pred} = x_{smooth} + \dot{x}_{smooth} * \tau$$

$$y_{pred} = y_{smooth} + \dot{y}_{smooth} * \tau$$

where the value of τ is computed as described below. Finally, the predicted values of ρ and θ corresponding to this position, required for target to track correlation, are determined by:

$$\rho_{pred} = \sqrt{x^2_{pred} + y^2_{pred} + h^2}$$

$$\theta_{pred} = \tan^{-1}(x_{pred}/y_{pred})$$

The values of $\dot{\rho}$ and $\dot{\theta}$ are not required unless the new ground range of the track places it sufficiently far from the sensor that ρ, θ tracking will be employed on the next scan. In that case, they are computed as follows:

$$\dot{\rho}_{pred} = \frac{x_{pred} * \dot{x}_{pred} + y_{pred} * \dot{y}_{pred}}{\rho_{pred}}$$

$$\dot{\theta}_{pred} = \frac{y_{pred} * \dot{x}_{pred} - x_{pred} * \dot{y}_{pred}}{\rho^2_{pred}}$$

and placed in the track file, replacing the values of $\dot{x}$ and $\dot{y}$.

When an aircraft is so close to the sensor that x, y tracking must be utilized, its time between reports can differ substantially from the scan period. The update interval, in fact, can vary from an arbitrarily small amount to one and a half times the scan period, as shown by Figure 9-7. To prevent unresolvable situations from occurring in target to track correlation, however, no update interval of less than half a scan will be permitted. If such a situation would occur, the update is delayed until the next aircraft report, as shown in part (a) of the figure.

To compute an accurate value for τ, the update interval, not only can the aircraft tangential velocity not be ignored, but it cannot even be assumed to be constant as was done above. Instead, the exact relationship shown in Figure 9-8 must be employed:

$$\arctan\left(\frac{x_o}{y_o}\right) + 2\pi(\tau-1) = \arctan\left(\frac{x_o + \dot{x}_o\tau}{y_o + \dot{y}_o\tau}\right)$$

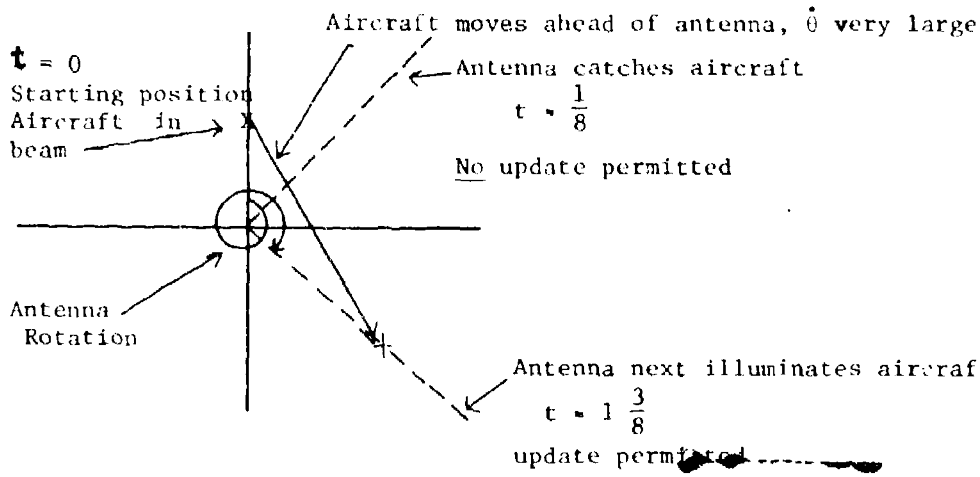
where x_o, y_o = smoothed position of current update

$\dot{x}_o$, $\dot{y}_o$ = track velocities

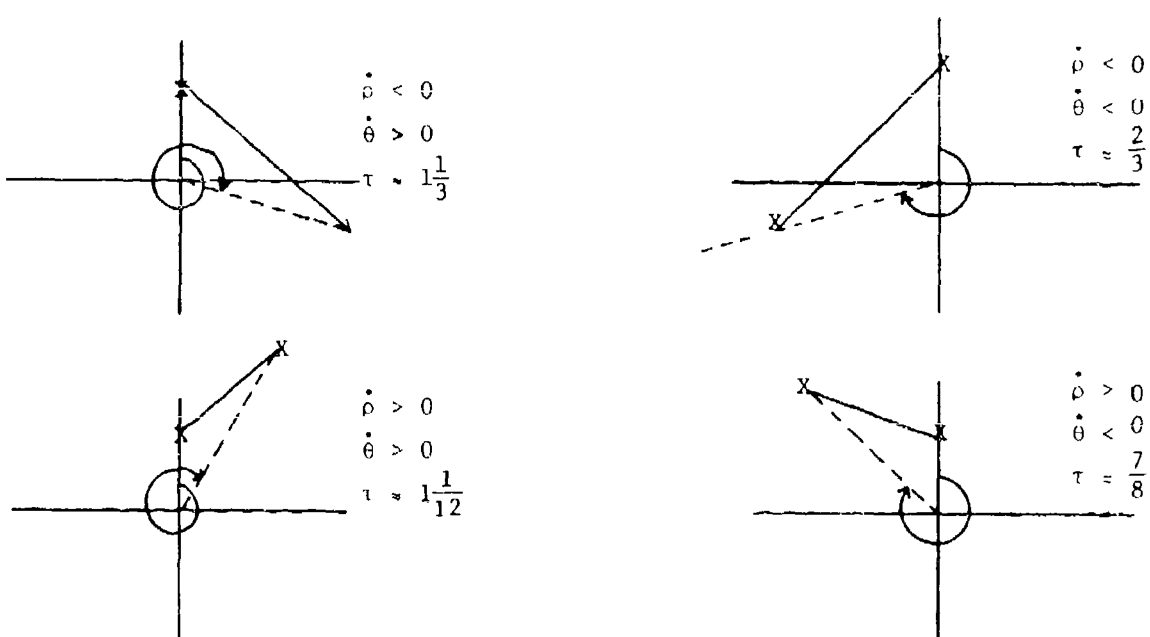
τ = update interval (to be found)

Simplifying this result:

$$\tan\left[2\pi(\tau-1)\right] = \tan\left[\arctan\left(\frac{x_o + \dot{x}_o\tau}{y_o + \dot{y}_o\tau}\right) - \arctan\left(\frac{x_o}{y_o}\right)\right]$$



(a) Special case, time between reports < $\frac{1}{2}$ scan

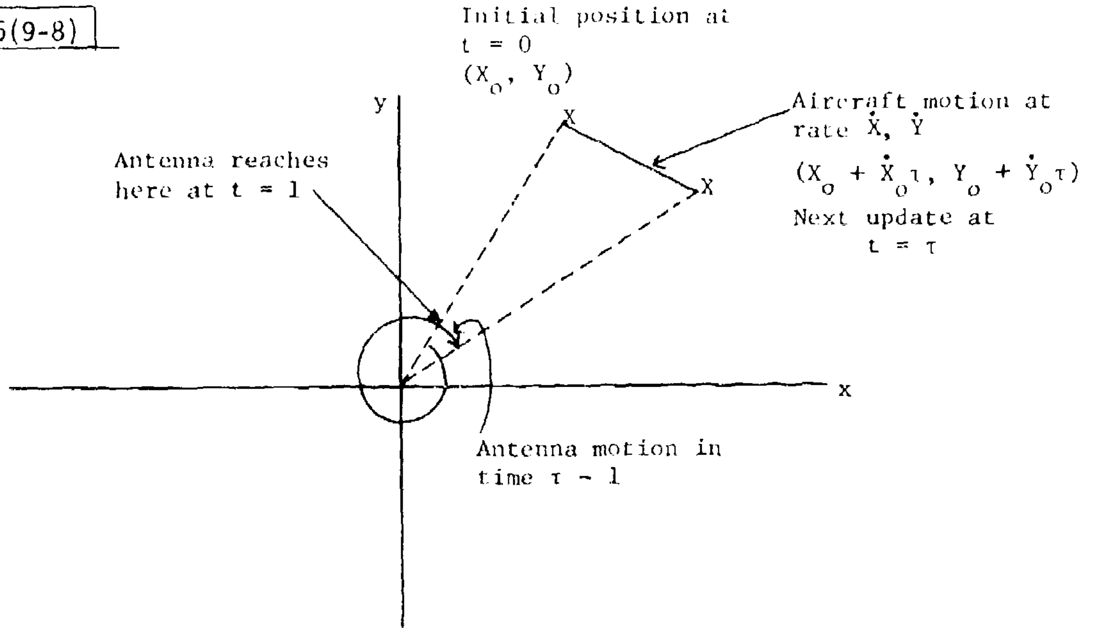


(b) Normal cases, $\frac{1}{2} < \tau < 1\frac{1}{2}$

ATC-65(9-7)

Figure 9-7: Possible Track Update Intervals

ATC-65(9-8)



Antenna rate is 2π radians/scan

Thus final angular position is $\tan^{-1}\left(\frac{X_o}{Y_o}\right) + 2\pi\ (\tau - 1)$

Final aircraft azimuth is given by:

$$\tan^{-1}\left(\frac{X_o + \dot{X}_o\ \tau}{Y_o + \dot{Y}_o\ \tau}\right)$$

Since antenna is illuminating aircraft at $t = \tau$:

$$\tan^{-1}\left(\frac{X_o}{Y_o}\right) + 2\pi\ (\tau - 1) = \tan^{-1}\left(\frac{X_o + \dot{X}_o\ \tau}{Y_o + \dot{Y}_o\ \tau}\right)$$

Figure 9-8: Exact Formula for Update Interval

$$\tan(2\pi\tau) = \frac{\dot{\theta}\tau}{1 + (\dot{\rho}/\rho)\tau}$$

where $\dot{\theta}$ and $\dot{\rho}$ are calculated as indicated earlier from x, y, $\dot{x}$, and $\dot{y}$. Since the tangent is a multiple valued function, the general solution must be written as:

$$\tau = \frac{R}{2} + \frac{1}{2\pi} \arctan\left(\frac{\dot{\theta}\tau}{1 + (\dot{\rho}/\rho)\tau}\right) \qquad R = 0, 1, \ldots$$

The correct value of R to use for a specific track can be determined by approximating τ. Once R is chosen, τ can be determined through iteration. Figure 9-9 shows how to determine R, while Figure 9-10 presents the detailed method for computing τ in all cases.

9.5 Updating a Coasted Track

If a track fails to receive a correlating target report on the current scan, it must either have its predicted position projected ahead to the next update time or be dropped from the system. The latter action is generally taken after a parametric number of successive correlation failures, although this rule is modified in two special cases.

The first special track drop situation pertains to tracks that were made part of a track grouping at initiation time (refer to Section 8.3). It should be recalled that only one track of such a group can correspond to a real aircraft. Thus, the following special track drop rule has been developed to eliminate as soon as possible the extraneous tracks:

- If a coasting track that has never correlated is part of a track grouping, and any other track in the grouping has successfully correlated, the coasting track is immediately dropped.

If a track in a grouping does correlate, the normal track drop rule will apply to it in the future.

The second special set of rules for track dropping apply to tracks whose predicted position lies within the sensor antenna cone of silence. In this region, defined as:

$$\rho_{pred} \leq h * \tan \theta_{cone}$$

where θ_{cone} is a parameter

$$\tau = \frac{R}{2} + \frac{1}{2\pi}\tan^{-1}\left(\frac{\dot{\theta}\tau}{1 - \tau/\lambda}\right) \qquad \text{where } \lambda = -\frac{\rho}{\dot{\rho}}$$

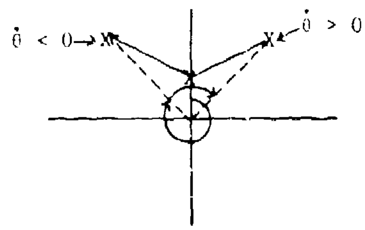
range is -1/4 to +1/4 (time until $\rho = 0$)

Thus:

R	Range of τ
1	$1/4 \rightarrow 3/4$
2	$3/4 \rightarrow 5/4$
3	$5/4 \rightarrow 7/4$

Case 1

$\lambda < 0$



$\dot{\theta} > 0 : 1 < \tau < \frac{5}{4}$

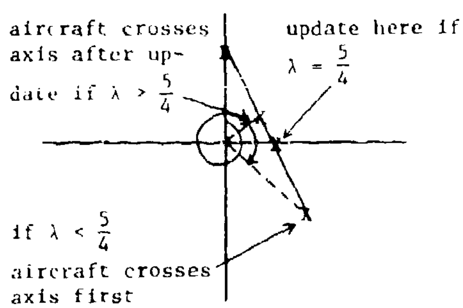
$\dot{\theta} < 0 : \frac{3}{4} < \tau < 1$

$\therefore$ R = 2 either sign

Case 2

$\lambda > 0$

$\dot{\theta} > 0$



$\lambda < \frac{5}{4} : \frac{5}{4} < \tau < \frac{3}{2}$

R = 3

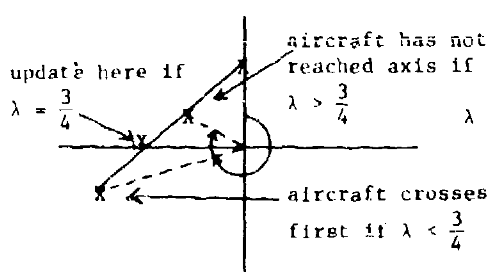
$\lambda > \frac{5}{4} : 1 < \tau < \frac{5}{4}$

R = 2

Case 3

$\lambda > 0$

$\dot{\theta} < 0$



$\lambda < \frac{3}{4} : \frac{1}{2} < \tau < \frac{3}{4}$

R = 1

$\lambda > \frac{3}{4} : \frac{3}{4} < \tau < 1$

R = 2

Figure 9-9: Choosing R for Update Formula

$$\tau = \frac{R}{2} + \frac{1}{2\pi}\tan^{-1}\left(\frac{\dot{\theta}\tau}{1 - \tau/\lambda}\right) \qquad \text{where } \lambda = -\frac{\rho}{\dot{\rho}}$$

ATC-65(9-10)

Solution Procedure:

1. Choose R (see Figure 9-9)
2. Choose τ_o
3. Iterate until τ converges

Special Cases: (no iteration required)

I.	$\dot{\theta} = 0,\ 0 \leq \lambda < \frac{1}{2}$	$\tau = \frac{1}{2}$
II.	$\dot{\theta} = 0,\ \frac{1}{2} \leq \lambda \leq 1$	$\tau = \frac{3}{2}$
III.	$\dot{\theta} \neq 0,\ \lambda > 1$ or $\lambda < 0$	$\tau = 1$

Usual Cases:

λ	$\dot{\theta}$	R	τ_o
< 0	any	2	1
$\geq \frac{5}{4}$	> 0	2	$1 + \frac{5/16}{\lambda}$
$< \frac{5}{4}$	> 0	3	$\frac{3}{2} - \frac{\lambda}{5}$
$\geq \frac{3}{4}$	< 0	2	$1 - \frac{3/16}{\lambda}$
$< \frac{3}{4}$	< 0	1	$\frac{1}{2} + \frac{\lambda}{3}$

Figure 9-10: Computation of Update Interval

no target reports are expected. Thus, to permit the track to coast through this region and be available for correlation when the aircraft reappears on the other side, the value of the track firmness f is not incremented. Since f cannot then reach the drop value, the track is kept in limbo. However, a count of the number of such scans is maintained in the track file, and this value is added to the firmness f to determine the size of the track's association zones. One additional condition is required for this rule to apply: the track must have a velocity of at least 50 knots. This insures that the track will eventually leave the cone of silence and not stay in the system in limbo forever.

Once the track exits the cone of silence, normal incrementing of f resumes if further track coasts occur. Should f then reach the proper value, the track is dropped. If the track correlates after leaving the cone of silence, however, the firmness is reset to 1 while the track history firmness g is set as follows:

$$g = \min \{f + \text{cone count}, \max f\}$$

where cone count = number of coasts in cone of silence

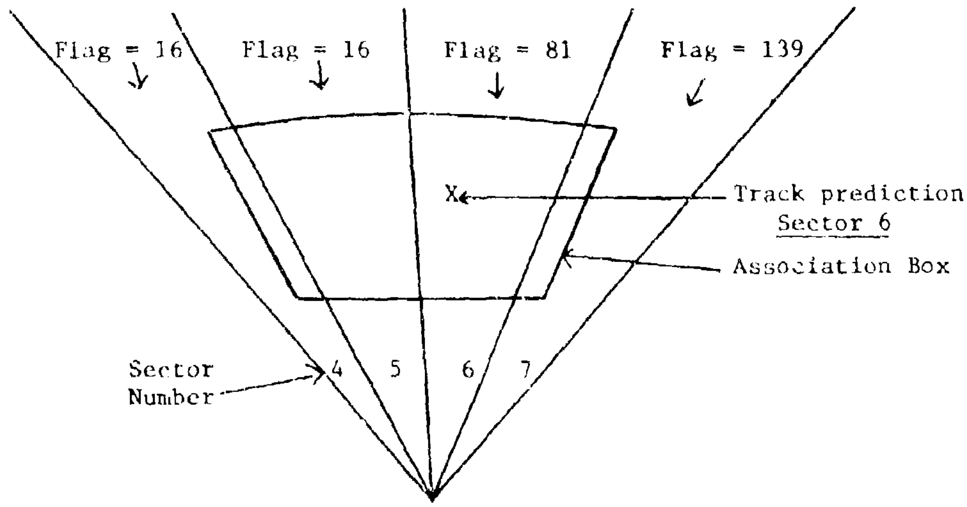
max f = maximum firmness value

If a track that has not correlated on the current scan is to be maintained in the system, its predicted position must be updated. Since no correlating report exists, no smoothing of the current predicted position is possible. Thus, only the projection step is performed for such tracks. The equations to use are identical to those for a correlated track, and employ ρ, θ, improved ρ, θ, or x, y coordinates depending upon the ground range of the track.

9.6 Sector Update

Every track resident on the current sector's list, whether or not its track file is being updated this sector, must be checked to determine in which sector it should next appear. The movement of tracks from sector to sector, as described in Chapter 7, is controlled by the flag variable associated with each track. The set of possible values for the flag, and the interpretation of each one, is presented in Figure 9-11.

To review, a track begins its activity in the first sector in which an associating target report could be found. The track then moves from sector to sector until it either finds a correlating report or reaches the last sector in which such a report could exist. When either event occurs, the target to track correlation process sets the track flag to zero. This setting signals track update that the time to process the track file has arrived.



Flag values given are those the track has when the corresponding sector is processed.

When the track correlates (or coasts), the flag is set to 0

Flag Setting (F)	Interpretation
F = 0	Update the track.
$10 < F \leq 74$	Track is predicted to be in sector F-10, which is subsequent to the current sector.
$74 < F \leq 138$	Track has already reached its predicted sector; the last sector for association is F-74.
F = 139	Track has already reached its last association sector.

Figure 9-11: Track Sector Flag Interpretation

For each track updated in the current sector, the program must determine the following three pieces of information:

1. The sector in which the track should first appear on the next scan
2. The flag setting the track should have at that time
3. Whether the track should be active or inactive

An active track in a sector is one that participates in the target to track correlation and track update processes, while an inactive track is ignored by both processes. This latter designation of track is required when a track is projected across a sector boundary. Assume, for example, that a track on the current scan is correlated in sector 4, while next scan it is predicted to be in sector 5. Hence, this track is immediately placed on the list for sector 5. If it were not made inactive, it would attempt to correlate again in the very next sector, or twice in one scan. By making it inactive, however, it is passed over until the next scan. All inactive tracks in a sector are converted to active status by track update after sector processing is concluded, which makes them available for correlation on the next scan.

The first sector in which an updated track can find an associating report on the next scan is determined by the extent of its zone 3 association box (refer to Section 6.2 for its definition). This box is bounded as follows (refer to Figure 9-12):

$$\rho_{min} = \left(\rho_{pred} - \rho^3\right) \leq \rho \leq \left(\rho_{pred} + \rho^3\right) = \rho_{max}$$

$$\theta_{min} = \left(\theta_{pred} - \theta^3\right) \leq \theta \leq \left(\theta_{pred} + \theta^3\right) = \theta_{max}$$

If ρ_{min} is less than zero, the association box covers the sensor, and an associating report could be found in any sector. To prevent unending searches, and unacceptably long data reporting delays, a parameter ΔS_{max} controls the number of sectors on either side of the predicted one in which a track may search. Thus, the first sector into which the newly updated track is placed is given by:

$$S_{first} = \text{Max} \left\{ S_{pred} - \Delta S_{max}, \frac{\theta_{min}}{\theta_{sector}} + 1 \right\}$$

where θ_{sector} is the azimuth extent of a sector and integer division is assumed.

ATC-65(9-12)

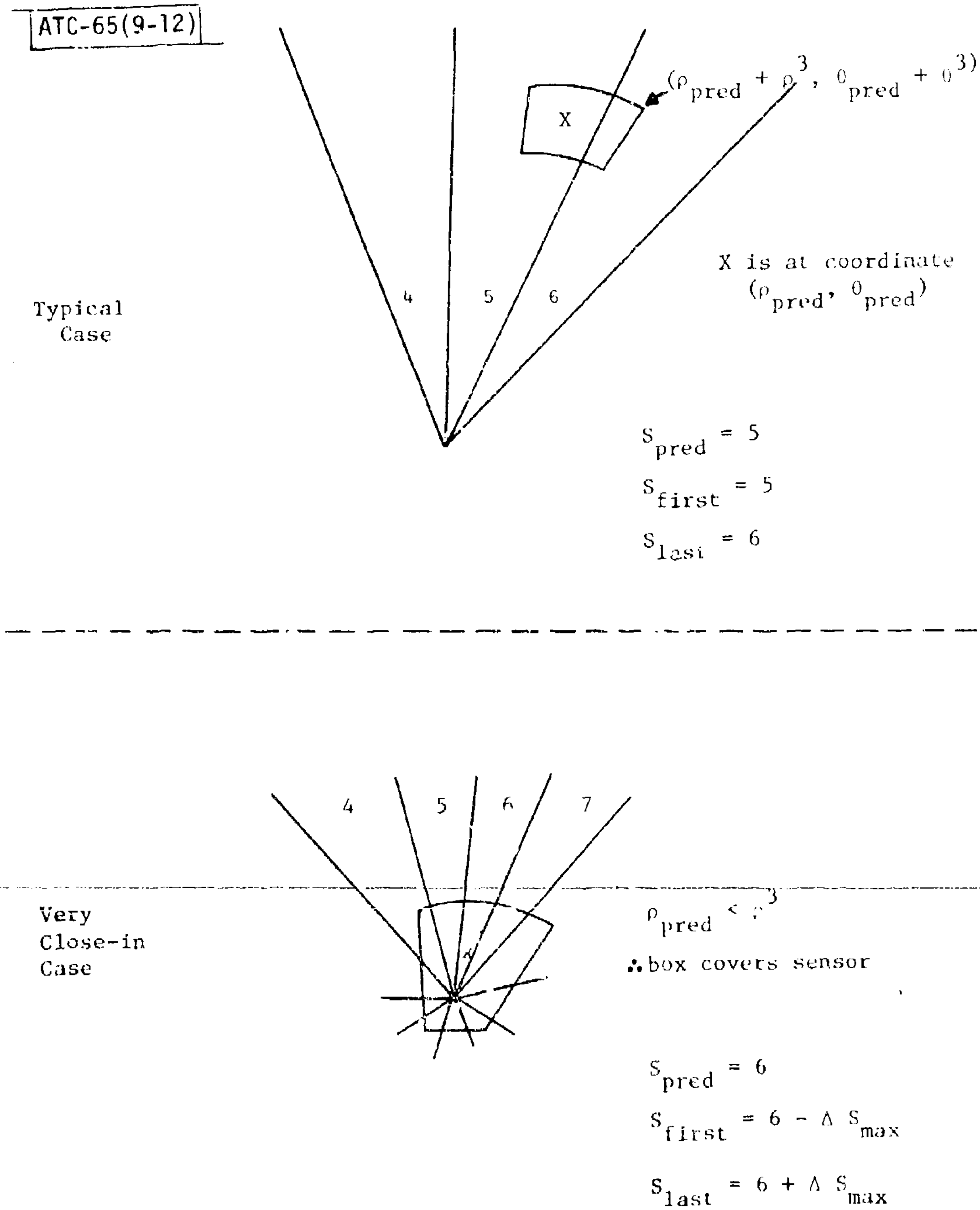


Figure 9-12: Extent of Association Sectors

This sector computation, and all others presented below, assumes arithmetic modulo N_{sector}, the number of sectors in a scan. For example, if $N_{sector}=32$, then a typical subtraction might be:

$$S_1 - S_2 =$$

$$4 - 27 =$$

$$-23_{32} = 9$$

which says that sector 4 comes 9 sectors later than sector 27. Similarly, sector 2 is "bigger" than sector 28 in the maximization.

If S_{first} is less than S_{pred}, which occurs when the association box crosses the left boundary of sector S_{pred}, the initial flag for the track becomes $10 + S_{pred}$, as indicated in Figure 9-11. No further computations are required.

If, however, the edge of the association box is contained within the predicted sector, which is the normal case for distant aircraft, the first sector and the predicted sector are one and the same. Then, the last sector in which the track can search must be calculated in order to set the track flag. This sector is given by:

$$S_{last} = \text{Min} \left\{ S_{pred} + \Delta S_{max}, \frac{\theta_{max}}{\theta_{sector}} + 1 \right\}$$

If S_{last} is larger than S_{pred}, the track flag is set to $74 + S_{last}$, while if the two sectors are identical, the flag setting becomes 139. This latter value is used to indicate that the track has reached the end of the line, and hence should be correlated at once.

Once the destination sector for the updated track is known, its status in that sector, active or inactive, can be determined. If the predicted next scan sector for the track, S_{pred}, preceeds or equals its current scan predicted sector, S'_{pred}, the track is automatically made active. Otherwise, if the track has moved clockwise, the number of sectors from the current sector to the next scan destination one must be computed:

$$\Delta S_{move} = N_S - (S_{curr} - S'_{pred}) - (S_{pred} - S_{first}) + (S_{pred} - S'_{pred})$$

where each parenthetical subtraction is modulo N_S. The rule then simply becomes: the track should be made active if $\Delta S_{move} \leq N_S$ and inactive if $\Delta S_{move} > N_S$.

Active tracks in the current sector that were not updated, because their correlation process is not yet completed, must be moved to the next sector so that the process can continue. It is vital that all such tracks be placed in order at the head of the list for the next sector (see Figure 9-13). Failure to observe this rule will lead to incorrect associations in that sector since the algorithm presented in Section 6-5 assumes no track can be further down in the list for one sector than it was in the previous sector.

Besides moving these tracks to the next sector, track update must compute the new flag settings for each such track. All cases that could arise are presented in Figure 9-14, along with the appropriate action to take. As is shown there, the flag setting changes only when the next sector is S_{pred} or S_{last} for the track. In the former case, S_{last} is computed and the track flag set as described above, while in the latter case the flag is set to 138. The sector S_{last} is given by:

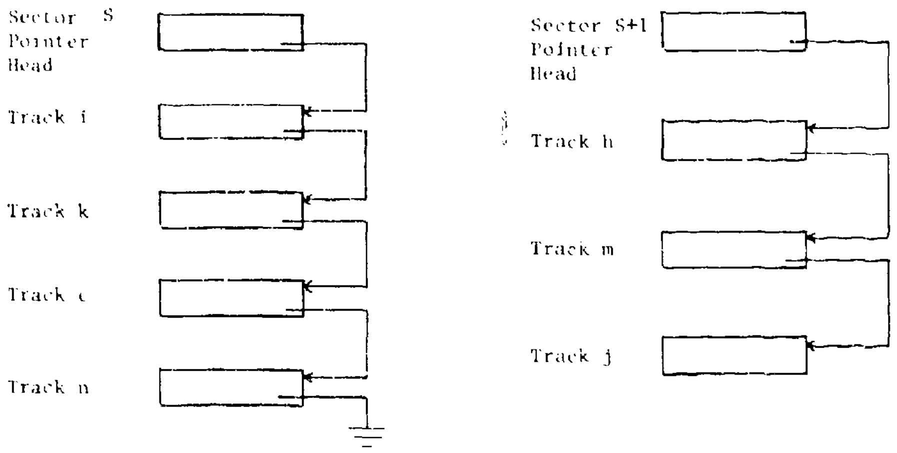
$$S_{last} = S_{pred} + \Delta S_{max} \qquad \text{if } \rho_{min} \le 0$$

$$= \text{Max} \left\{ S_{pred} + \Delta S_{max}, \frac{\theta_{max}}{\theta_{sector}} + 1 \right\} \qquad \text{otherwise}$$

All tracks moved to the next sector are automatically made active.

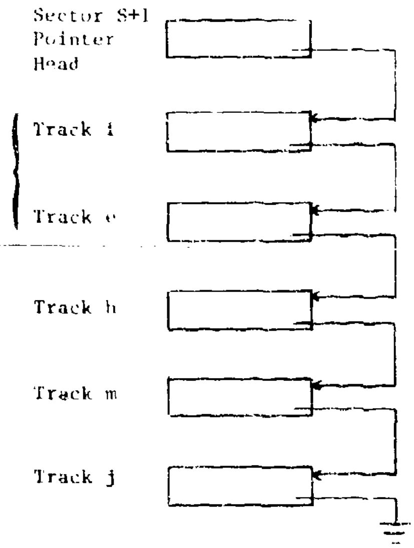
The final action of track update, as mentioned earlier, is to convert the status of all inactive tracks in the sector to active. These tracks are left in the list for the current sector, and their flag settings are not changed. Then, when the same sector arrives on the next scan, they are ready to begin the association, correlation, and track update sequence.

Assume the following sector linkages exist before Section S is processed:



Then, if tracks i and e are both moved to the next sector, the sector linkage for Section S+1 becomes:

Moved tracks are placed at <u>head</u> of list, and in <u>same order</u> (i before e) as in previous sector.



<u>Figure 9-13:</u> Track Moving Rule

Current Sector (S) Flag	Flag Assigned for Next Sector (S + 1)	
10 + (S + 1)	$\begin{cases} 74 + S_{last} \text{ if } S_{last} \neq S + 1 \\ 139 \text{ if } S_{last} = S + 1 \end{cases}$	
10 + (S + i), i > 1	10 + (S + i)	No change
74 + (S + 1)	139	
74 + (S + i), i > 1	74 + (S + i)	No change
139	139	No change

ATC-65(9-14)

Figure 9-14: Track Flag Update Rules

10.0 FALSE ALARM TARGET REPORTS

It is unfortunate that in any ATCRBS system some of the target reports created by reply correlation do not correspond to the position of real aircraft. The main categories of such false alarm reports are:

1. False targets

2. Fruit targets

3. Split targets

4. Ringaround targets

Various algorithms are included in the surveillance processing functions which attempt to either mark these reports as false or eliminate them from the system output stream. This chapter will discuss both the identification and disposition algorithms for each type of false alarm report.

False targets are generally caused by the reflection of aircraft responses off buildings, hangars, or other structures near the sensor, thereby causing an apparent aircraft position behind the reflector. Depending upon the size of the reflector, such false targets may persist for several scans and initiate false tracks. Since the reflection mechanism is deterministic, it is possible, given the reflecting surface parameters, to compute the position of the aircraft whose signal was responsible for the false target. If a track exists near this calculated position whose identity code and altitude agrees with the potential false target, it is reasonable to conclude that the report is indeed false.

One other type of false target that is identified by surveillance processing is that due to ground reflection. This mechanism produces two reports at about the same azimuth, one (the reflected one) at greater range than the other. The discrete correlation process declares such a situation whenever two reports with the same discrete code are found in the same sector (refer to Chapter 5). Cases of non-discrete ground reflection false targets can not be identified in this system as two aircraft with the same non-discrete code in the same azimuth sector are quite common.

A fruit reply results when an aircraft reply sent in response to an interrogation from another sensor is received at the local sensor. Since the interrogation times of the two sensors are different, the local sensor will compute an incorrect range for the aircraft based on the assumed turn-around time from its own interrogation time. By design, the repetition rate of any two sensors in an area is different, and thus successive fruit replies from the same aircraft due to the same interrogator will not agree on range when processed by the local sensor, and thus not be correlated.

However, it is possible for two fruit replies from different aircraft, or two fruit replies from the same aircraft due to different interrogators, to coincidentally agree on range and azimuth and thus produce a fruit report. Generally, such a fruit report will not correlate with an existing track, and will consist of two replies, one of mode A and one of mode C. Thus, it is often possible for surveillance processing to identify and discard fruit target reports.

The third type of false alarm report, a split, occurs when the reply sequence from an aircraft is separated by reply correlation into two or more target reports. This can result from code or azimuth declaration errors in the reply processor, from intermode delay variations in aircraft transponders, or from various environment effects. Many of the more common types of splits have easily recognized characteristics that permit them to be identified and then discarded. Finally, ringaround target reports are defined as those formed by high elevation angle, short range sidelobe replies which are not flagged as sidelobe because of the failure of the antenna patterns in that region. As with other false alarm reports, ringaround reports have identifiable characteristics that can lead to their discovery and elimination.

10.1 False Target Identification Process

The geometrical situation that exists when a false target is produced is depicted in Figure 10-1. The angle θ and range ρ are contained in the suspect target report, while the reflector distance d and orientation angle ϕ are parameters that have been fed into the surveillance processing program. The unknown values that must be calculated are thus the range ρ' and azimuth θ' of the aircraft generating the false target. If a track is found near that location that agrees on code and altitude with the suspect report, the report can reasonably be labelled false.

In order to standardize the computation of ρ' and θ', all candidate false target situations are rotated into the first quadrant. The conversions required for each quadrant shift are:

$$\hat{\theta} = \theta - 90^{\circ}$$

$$\hat{\phi} = \phi - 90^{\circ};\ \text{if } \hat{\phi} < 180^{\circ},\ \hat{\phi} = \hat{\phi} + 180^{\circ}$$

where the second step for $\hat{\phi}$ guarantees that $180^{\circ} < \hat{\phi} < 360^{\circ}$ as required for the computations. The set of equations that are used to compute θ' and ρ' are presented in Figure 10-2.

Once the position of the alleged real aircraft has been found, the next step is to examine the existing system tracks located near that spot to determine whether there is one that matches the suspect report in code and altitude. In order to simplify this search, all real tracks are maintained in

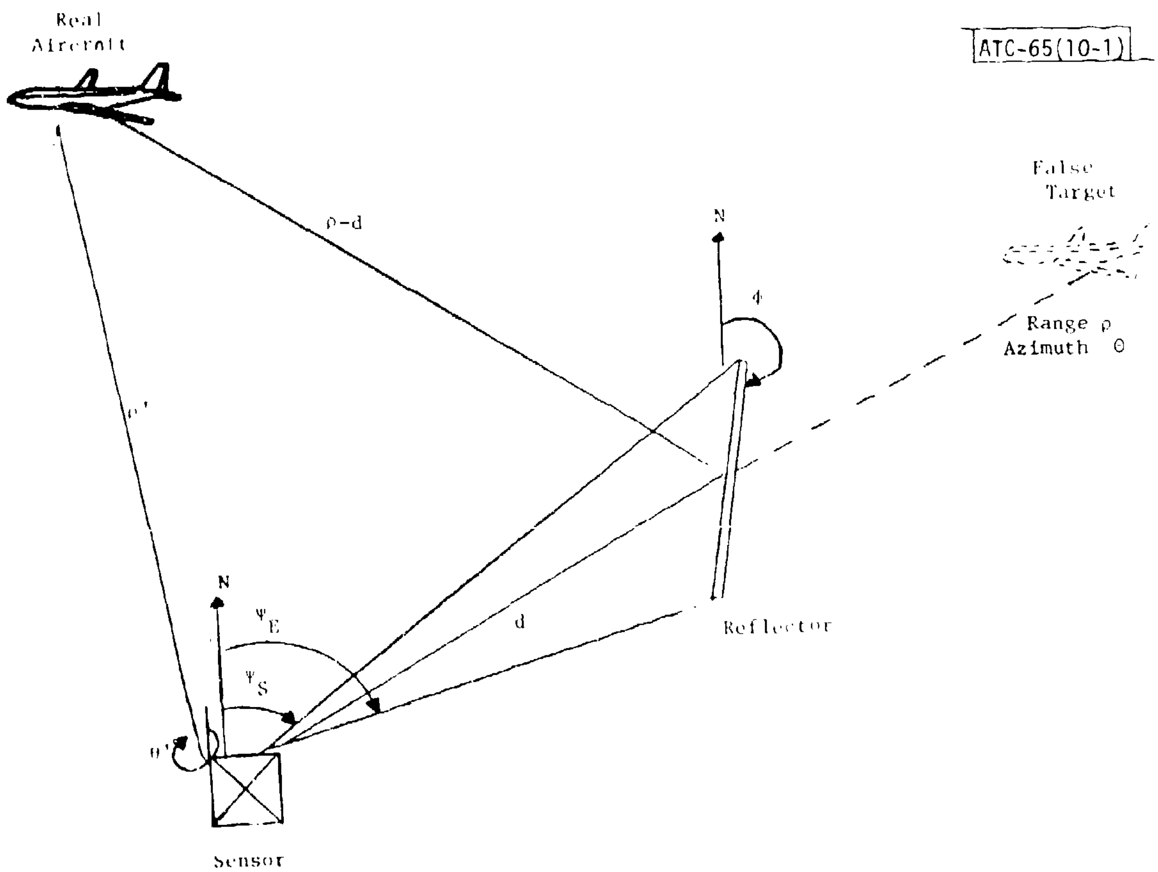
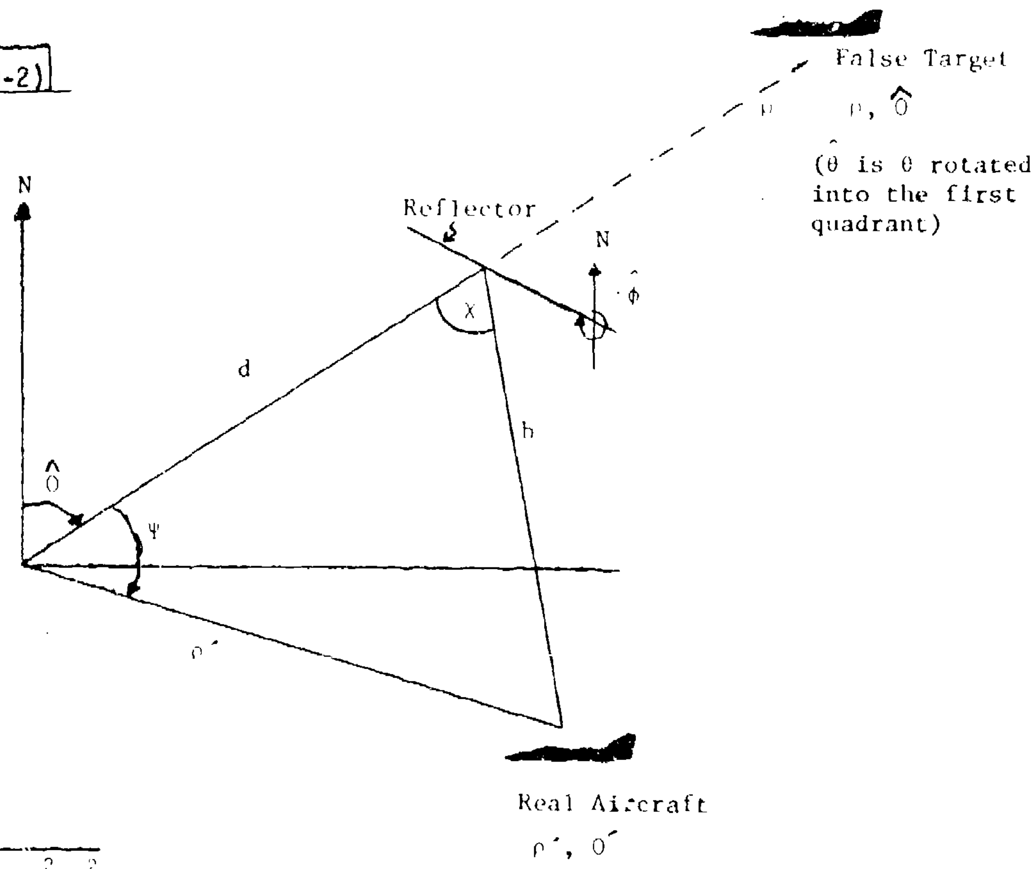


Figure 10-1: False Target Geometry



$b = \sqrt{(\rho - d)^2 - h^2}$ (if argument < 0, target is real)

$\chi = 540^{\circ} - 2\left[\hat{\phi} - \hat{\theta}\right]$

$\rho' = \sqrt{b^2 + d^2 - 2bd\cos\chi}$

$\Psi = \sin^{-1}\left[\frac{b \sin\chi}{\rho'}\right]$

if $b^2 > (\rho')^2 + d^2$, $\Psi = \begin{cases} 180 - \Psi & \text{if } \Psi > 0 \\ -(180 - \Psi) & \text{if } \Psi < 0 \end{cases}$

$\theta' = \theta + \Psi$

Figure 10-2: Computation of Real Aircraft Position

a range sort table. This table, illustrated in Figure 10-3, uses one word per bin to identify the first track in the bin, and then links together all subsequent tracks in that bin through pointers. Every time a new track is initiated, an entry is created for it in the proper bin, determined by:

$$b = \frac{\rho_{gnd}}{\Delta\rho_{bin}} + 1 \text{ (integer division)}$$

where ρ_{gnd} is the predicted ground range of the track and $\Delta\rho_{bin}$ is the extent of a sort bin. Thereafter, each time the track is updated, its new and old predicted ground ranges are compared. If both values map into the same bin, no action is taken; otherwise, the previous track entry is deleted and a new one is created. The old and new ground ranges determine the two bins affected. Finally, when a track is dropped, its entry is removed from the table.

Ideally, if the report is indeed false, a track will be found whose position is very close to the calculated point and whose code and altitude agree perfectly with the report. Unfortunately, this ideal state is often not encountered. Since no reflecting surface is perfectly flat, the computed position could be significantly in error. Also, the track will never perfectly represent the location of the aircraft at the time of the reflection. Thus, fairly substantial positional deviations between the computed point and the track prediction can exist. In addition, no surface is uniformly reflecting. Thus, one or more bit differences could exist between the code or altitude of the report and that of the real aircraft. In some cases, in fact, only one mode of reply may be reflected. Thus, imperfect code or altitude matches may exist between the target and the track.

It is clear then that a problem exists in the matching part of the algorithm. If too tight a match is required between candidate report and existing track, actual false targets would often be called real. On the other hand, too loose a match could result in real targets being labelled false. This problem has been resolved by defining two sets of match criteria.

A candidate target will be called <u>false</u> if a track is found that satisfies all of the following tight conditions, where position is relative to the computed aircraft point and code and altitude are relative to the report itself:

(a) $\Delta\rho \leq \delta\rho_{tight}$

(b) $\Delta\theta \leq \delta\theta_{tight}$

(c) $\Delta C = 0$

(d) $\Delta h \leq \frac{1}{2}\,\Delta h_{max}$

ΔC and Δh are computed in the same manner as for target to track association (refer to Section 6.2).

ATC-65(10-3)

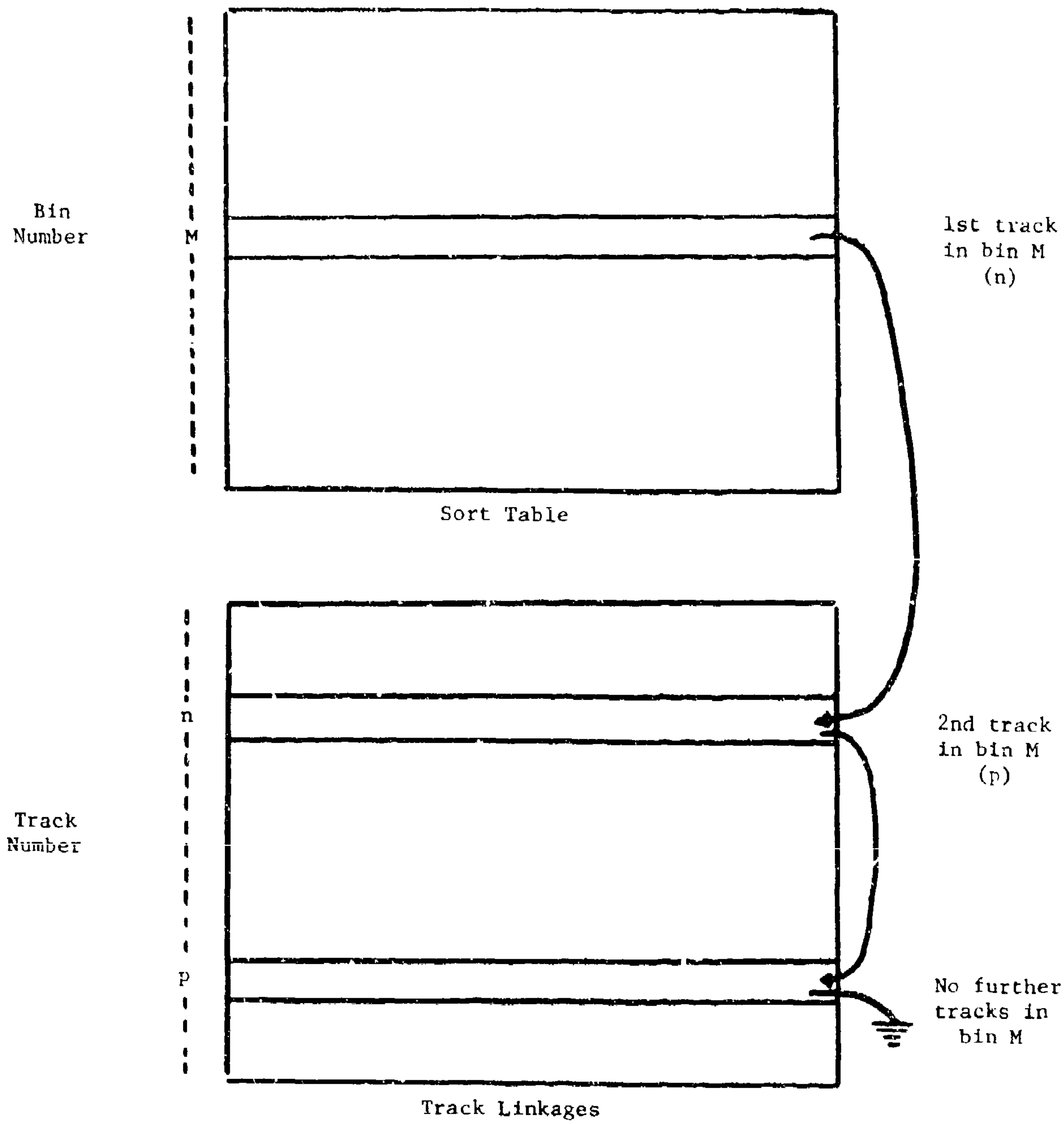


Figure 10-3: False target sort table.

If the report has a discrete 4096 code, and a track is found with the same discrete code, a higher degree of confidence exists that the target is indeed false. Thus, for this special case, the set of false conditions is loosened as follows:

(a) $\Delta\rho \leq \delta\rho_{loose}$

(b) $\Delta\theta \leq \delta\theta_{loose}$

(c) same discrete code

(d) $\Delta h \leq \Delta h_{max}$

The second type of match that can occur is called possibly false. This occurs when no track is found for the report that satisfies the false conditions listed above, but a track exists that meets the following looser conditions relative to the report:

(a) $\Delta\rho \leq \delta\rho_{loose}$

(b) $\Delta\theta \leq \delta\theta_{loose}$

(c) $\Delta C \leq \Delta C_{max}$

(d) $\Delta h \leq \Delta h_{max}$

The interpretation and use of targets labelled possibly false will be given in the next section.

10.2 False Track Algorithm

False targets, particularly those due to major reflectors, tend to persist for a large number of scans. This fact, combined with the difficulty of positively identifying many false targets, creates a problem for the system. If false targets were simply eliminated when found, targets would tend to flicker on and off the controller's screen during a false target sequence. In a particularly bad case, several real tracks could even start and drop during the sequence. This would occur because those targets that were not positively identified as false would have to be called real, used in surveillance processing, and output to ATC.

In order to prevent such situations, false target reports are specially marked when identified but are not eliminated from the surveillance processing algorithms. Instead, these reports are permitted to initiate false tracks and to correlate to existing ones. Then, should an unsure report correlate

to a false track, the report may be labelled false with reasonable certainty. Hopefully, this use of past knowledge will result in all reports produced during a false target sequence being labelled as false.

If a false track were incorrectly called real, some inconvenience might result if pilots were ordered to avoid a nonexistent aircraft. On the other hand, if a real track were to be labelled false, a catastrophic collision could occur. Thus, whenever uncertainty exists in the status of a track, it will be labelled real to the ATC facilities. In addition, once a track is called real by surveillance processing (as opposed to uncertain), it will not be permitted to convert to false at any future time. This latter condition, in addition to providing system safety, helps to cut down considerably the execution time of the system; over 90% of all targets will correlate with real tracks, and by this rule, none of these need enter into the complex false target identification process.

The reports, then, that must be checked for falseness fall into two categories: those that are uncorrelated, and those that correlate to tracks not called real (i.e., false or possibly false tracks). The false target identification test for these reports consists of two parts: the zone test and the image test. Reports that fail the zone test are labelled real, while those passing it enter the image test for final status determination.

The zone test checks to see whether or not the candidate report is in an azimuth wedge that corresponds to a known reflector. In order to permit this decision to be made reasonably quickly, the reflectors specified for each site are azimuth ordered. Furthermore, the number of the first reflector located in each sector (either totally within or straddling the boundary) is kept in an array. With this implementation, the zone test consists of comparing the report azimuth with the beginning and ending azimuth of each reflector in the sector, starting with the known first one. If the report azimuth falls within the reflector wedge, the test is passed; if the report azimuth is less than the starting azimuth of the reflector, the test is failed (due to reflector ordering); otherwise, the next reflector is considered and the test continues.

Targets passing the zone test are next subjected to the image test. This test, presented in detail in the previous section, seeks to locate the track corresponding to the aircraft that produced the target report if it were indeed due to a reflection off the surface identified during the zone test. The result of this test will be that the candidate report is declared to be real, false, or possibly false. Refer to the previous section for the criteria used for this decision.

Since the image test is searching for a track, a complication can arise if the false targets and real targets due to an aircraft begin on the same scan or adjacent scans. In such a situation, the first false target would have to be labelled real, as no track would yet exist for the aircraft. To

prevent such an incorrect decision, the following modification has been adopted: no uncorrelated report that passes the zone test can be called real due to failing the image test; instead, it is labelled possibly false. Should such a report initiate a track, and a report that correlates to this track be labelled real, the decision is accepted and the track called real. By this time, of course, the real track for the aircraft would already exist and failure of the image test would constitute acceptable proof.

Surveillance processing recognizes four modes of tracks with respect to falseness: real, possibly false type I, possibly false type II, and false. The state diagram that defines these categories is presented in Figure 10-4. The circles represent the modes, while the arrows specify the transitions that occur when the status of the correlating target reports are determined. For example, a possibly false type I track that correlated with a false target becomes possibly false type II. An examination of the diagram reveals that the following rules apply:

1. A track that is initiated with a real report, or ever corelates to a real report, is real forever after.

2. A track is false only if two or more of its reports (initiation ones or correlating ones) are definitely declared to be false.

3. Until a track is declared false, possibly false reports merely prolong the final decision.

To the outside world, a possibly false track and its correlating reports are both labelled real. Thus, the possibly false category serves as a holding action by permitting a track to eventually be labelled false when enough evidence is gathered. If this category did not exist, suspect reports would have to be called real, and hence many false tracks would be mislabelled. One modification to the state diagram should be mentioned: if a track is still in a possibly false state after 10 reports, it is converted to real. This is done to prevent a track being followed by ATC from suddenly dropping out of sight.

For the most part, false tracks are processed exactly the same as real tracks by the correlation algorithms. The main difference, of course, is that reports correlating to false or possibly false tracks must be checked by the false target routine. One other modification has been found necessary however. False target sequences tend to end in the middle of the coverage region, as opposed to at long range or at airports like real report sequences. Thus false tracks, while they are dropping, are ripe to correlate with extraneous reports of all types. To prevent the resultant clutter from interfering with ATC, these correlations should be suppressed. The following rule attempts to implement this desire: if a false track is to be correlated with a target called real, and the track and target codes disagree (i.e., $\Delta C > \Delta C_{max}$), the correlation is rejected and the report is treated as uncorrelated. This rule has proven itself empirically.

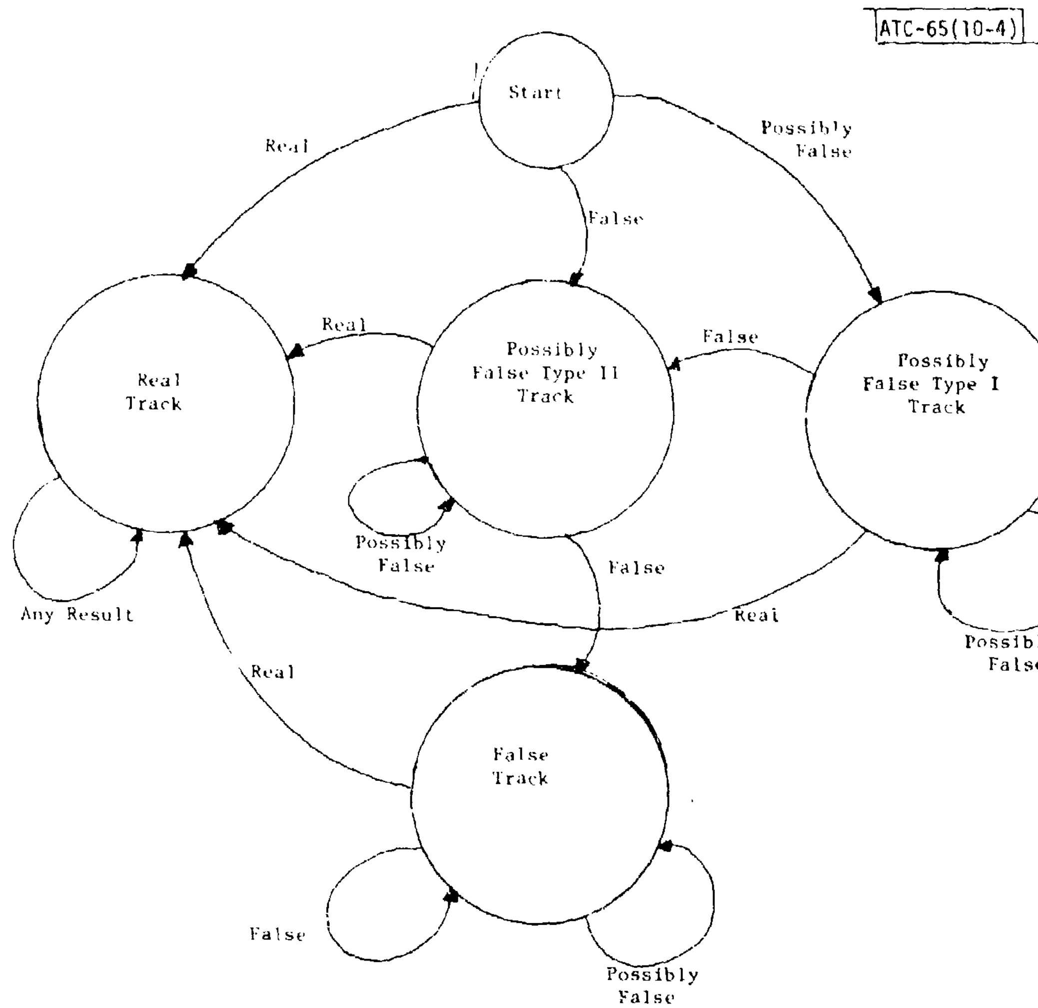


Figure 10-4: False Track State Diagram

10.3 Fruit Reports

The second category of false alarm report is that caused by fruit replies. Generally, the minimum number of replies in a report is set at two (with only mode A and C replies being counted), so that fruit reports occur when two or more fruit replies coincidentally correlate. However, one possible mode of operation for the ATCRBS system is to declare even uncorrelated replies to be valid target reports. This mode would be employed, of course, only where fruit levels were extremely low.

Even if a sensor is located in such a benign environment that uncorrelated replies are declared as reports to improve round reliability during fades, the large majority of such replies will still be fruit. Thus, to prevent these replies from causing tracking errors, 1-hit reports are treated with suspicion in several places in surveillance processing. In particular, the following actions described elsewhere in this paper fall into this class:

1. The association zone of a 1-hit report association is increased by one over the calculated value (and thus a 1-hit report falling in a track's box 3 is rejected).

2. 1-hit report associations receive worse Quality Scores than multiple hit ones.

3. A 1-hit report is not permitted to correlate with a not yet established track (i.e., one who has not yet existed for 5 scans).

4. An uncorrelated 1-hit report is dropped from the system, and so is not used in track initiation or output to ATC.

The first two penalties insure that 1-hit reports are not used to update a track unless they provide a good match for the track and no reasonable multiple hit report is available. The third rule attempts to insure that extraneous tracks are not kept alive by fruit reports. The final rule guarantees that 1-hit reports not used to update tracks in fades cannot cause any harm to the system.

In the normal mode of operation, with 1-hit reports suppressed, fruit targets are formed only when two or more fruit replies correlate with each other. Since, by system design, fruit correlation is a random event, hardly ever will a fruit report contain more than two replies. Although the number of fruit targets per scan is dependent upon the environment and the sensor parameters, experience has shown that about 1-2% of all reports declared fall into this category.

In order for two fruit replies to correlate, they must closely agree on range and azimuth. In addition, if the replies are of the same mode, they must agree on code. Thus, since code agreement is unlikely, most fruit reports will consist of one reply of each mode. Furthermore, the most likely

sweeps on which to find a correlating reply for a fruit are those adjacent to its sweep (due to the azimuth correlation requirement). Since adjacent non-mode 2 sweeps are of opposite mode, this reinforces the conclusion that fruit reports are of type A/C. The actual fraction of all fruit reports that are A/C is given by:

$$1 - (1 - \frac{2}{N}) P_M$$

where N is the runlength and P_M is the probability of code agreement. Thus, for typical values, over 90% of all fruit reports have one reply of each mode.

The method required to eliminate fruit targets is thus quite obvious. If a report consists of 1 mode A and 1 mode C reply (the number of mode 2 replies irrelevent), and fails to correlate with an existing track, it should be deleted from the system. It should also be remembered that A/C reports are penalized in both the discrete and non-discrete correl·tion algorithms relative to multiple hit reports. If radar information is available to the system, this requirement is altered by adding "and not radar reinforced" to the condition. The only system drawback to this policy is that on occasion tracks will require more scans to be initiated, as valid reports are discarded. However, studies have shown this effect to be unimportant.

Of the remaining fruit reports, namely those with two replies of the same mode, about half are mode A only and half mode C only. Targets with only mode A replies are generally due to aircraft without mode C responding capability. Thus, such reports cannot be eliminated as fruit. Targets with only mode C replies, however, are virtually never due to real aircraft. Thus, reports of this type should also be eliminated when uncorrelated (and unreinforced).

10.4 Split Reports

In theory, all replies from the same aircraft will be declared with about the same range and azimuth, and all replies of the same mode with the same code. In practice, however, various system defects can cause some replies of a sequence to be declared incorrectly. When such an event occurs, the reply correlation process will split the replies from an aircraft into two target reports. This section will review the various methods that surveillance processing uses to identify and eliminate various types of splits.

Hardly ever does the ATCRBS reply processor make an error in determining a reply range. Thus, almost all range splits are caused by improper transponder turn-around delays. The only such delay error that leads to range splits rather than constant bias errors is an out-of-spec intermode delay variation. Such an occurrence will lead to mode A replies having a different

perceived range than mode C replies. Then two reports will be declared, each containing only one mode. The first function of surveillance processing, as explained in Section 5.1, is to search for pairs of such single mode reports that correlate on azimuth. Whenever a pair is found, the two reports are reconstructed into one.

There are two mechanisms that can cause the reply processor to declare some replies in a sequence with the improper azimuth, one random and one systematic. Random azimuth errors occur when interference on the reference pulse causes the monopulse to be read incorrectly. Since the effect is to produce a random value, the reply in question will generally not correlate with any other reply and hence be eliminated as a fruit.

Systematic azimuth errors, usually called "tailing", occur when the monopulse calibration curve does not match the reply characteristics of a particular aircraft. This can occur for example when the aircraft frequency, amplitude, or elevation angle is unusual. The effect is that replies at one edge of the beam may fail to correlate with these in the center or other edge. If tailing causes one reply to not correlate, it will be eliminated as fruit. If two successive replies correlate with each other but not with the remainder, they will form a 2-hit A/C report which will be eliminated as a fruit report (as described in Section 10.3). No case of tailing ever encountered has resulted in the creation of two reports, each having three or more replies.

In order for two replies of the same mode to correlate, they must agree in all mutually high confidence bits. Thus if the reply processor makes a high confidence bit error due to any of a large number of low probability effects, two reports will be created for the aircraft. During the reply correlation process, an attempt will be made to correlate replies of the second group with those of the first. Although the attempt will fail due to the code difference, the range and azimuth tests will be passed. This will result, as explained in Section 4.6, in each report being marked as a code swap candidate. If the code swap occurs during association, the losing report is eliminated. Even if no code swap is required, if one of a pair of swap candidate reports is correlated and the other fails to correlate, the latter is eliminated as a code split during track initiation. Thus, only if a code split occurs during the first two scans of an aircraft's life will it not be rectified.

10.5 Ringaround Reports

A sensor antenna, being highly directional in nature, transmits most of its interrogation energy through its narrow mainbeam. However, an aircraft sufficiently close to the sensor, even though it is located in an antenna sidelobe, can still receive enough energy from an interrogation to pass its transponder threshold. Furthermore, were such an aircraft to respond to the sidelobe interrogation, its reply, even though received through the same sidelobe, would be strong enough to pass the sensor threshold. Such responses, if left unchecked, would of course lead to numerous spurious target reports.

To combat the occurrence and acceptance of sidelobe replies, a sensor is equipped with an omni antenna. An aircraft can then distinguish mainbeam interrogations from sidelobe ones by noting whether a stronger signal is received from the directional or omni antenna respectively. Similarly, a sensor can filter out sidelobe replies by ignoring those replies received more strongly by the omni antenna. Thus, aircraft can be prevented from responding to sidelobe interrogations, and sensors can eliminate sidelobe responses (mainly fruit from aircraft in the mainbeam of other sensors).

Various system effects, particularly the failure of the omni and directional antenna patterns to track each other at high elevation angles, can cause this mechanism to fail. When such a case arises, replies from an aircraft will be accepted over a wide azimuth extent. Since all replies are mapped into a small azimuth wedge centered at the antenna boresite, the result will be a number of target reports at the same range scattered over the azimuth acceptance region. Figure 10-5 illustrates this effect and the resulting report pattern. This phenomenon, because of its characteristic appearance on a radar scope, is known as ring-around.

From this description of ring-around, it is clear that the extraneous targets generally possess the following properties:

1. They fail to correlate with a real track

2. They are at short range

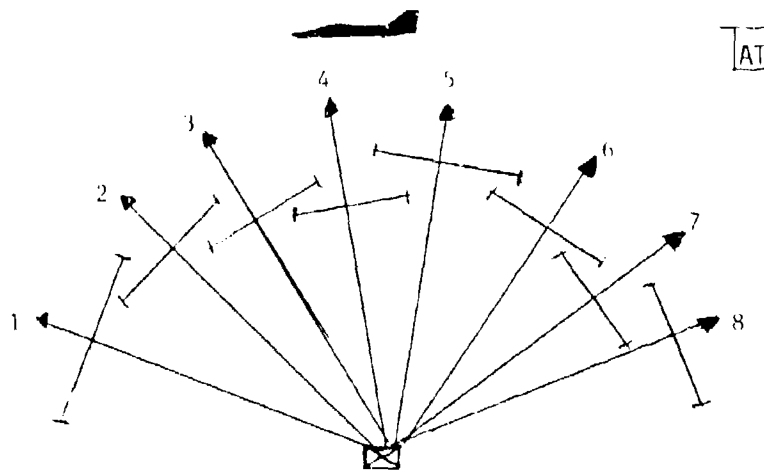
3. They have a high elevation angle

4. There is a real track with the same code and altitude at approximately the same range

Surveillance processing takes advantage of these unique characteristics to mark all such targets as false. The algorithm that accomplishes this has two parts: screening and matching. The screening section checks a report to see whether it meets the first three properties listed above using parametric range and elevation cutoffs. For reports without known altitude, the elevation test is bypassed. Also, if the report correlates with a false track, such as one started by previous scans ring-around, it is still acceptable.

The matching part of the algorithm attempts to locate a real track to which reports passing the screening could correspond. The process used is simply a subset of the false target algorithm presented in Section 10.1. The "reflecting surface" is taken to be the sensor, and all orientation angles are assumed. This latter assumption effectively disables the azimuth correlation requirement. The remainder of the identification process is identical to the false target image test. Also, tracks initiated by ring-around reports are labelled and processed identically to the false tracks described in Section 10.2.

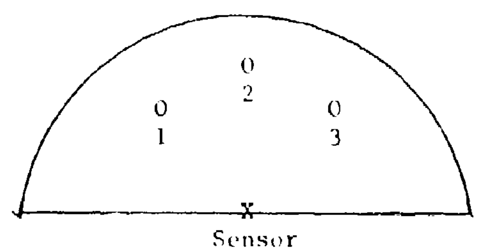
ATC-65(10-5)



Arrow indicates boresight direction

indicates extent of azimuths that can be produced on the sweep

Sweep	Mainbeam?	Correlates With	
1	no	-	tgt 1
2	no	1	
3	yes	-	tgt 2
4	yes	3	
5	yes	3	
6	no	-	tgt 3
7	no	6	
8	no	6	



display on controller's scope

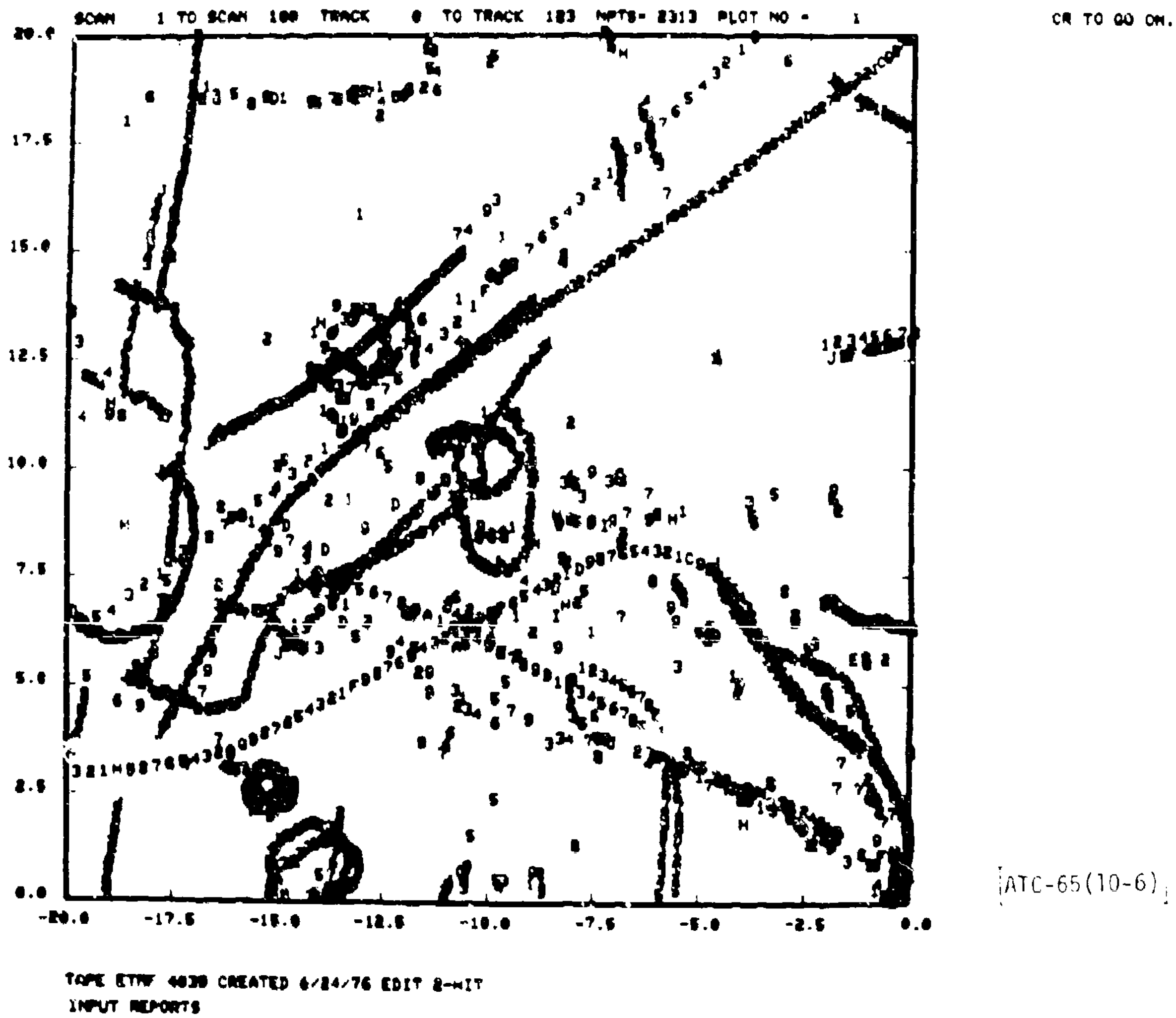
Figure 10-5: Ringaround Effect

10.6 Data Editing Example

This section presents an example of how effectively the data editing routines described in this chapter work on real data. The data employed was collected at Washington National Airport by the Transportable Measurement Facility (TMF).

Figure 10-6 displays all target reports declared by the reply processor over a period of 100 scans for a particular area of the overall coverage region. Clearly, numerous extraneous reports are seen to be cluttering up the picture. If no data editing were applied, the correlated reports that would have resulted from this input are shown by Figure 10-7. Although this picture is a major improvement, a large number of false alarm tracks are apparent.

Next the same input data was processed with the data editing routines enabled. The first step of data editing is to identify and eliminate fruit, split, and sidelobe reports. Figure 10-8 demonstrates the number of such extraneous reports that were found. Next, false targets are located and marked. Figure 10-9 illustrates how many of these were found to be present, while Figure 10-10 shows the false tracks they initiated. When both sets of reports are deleted, the set of reports remaining are the ones believed to be valid. Figure 10-11 depicts these reports. Comparing this figure with 10-6, it is clear that a tremendous improvement has been made in the output data quality. Finally, Figure 10-12 presents the valid, correlated reports. If these are the only reports used by ATC, as we recommend, it is obvious that the effect of false alarm reports will be very minimal in the air traffic control system.



ATC-65(10-6)

Figure 10-6: Input Reports

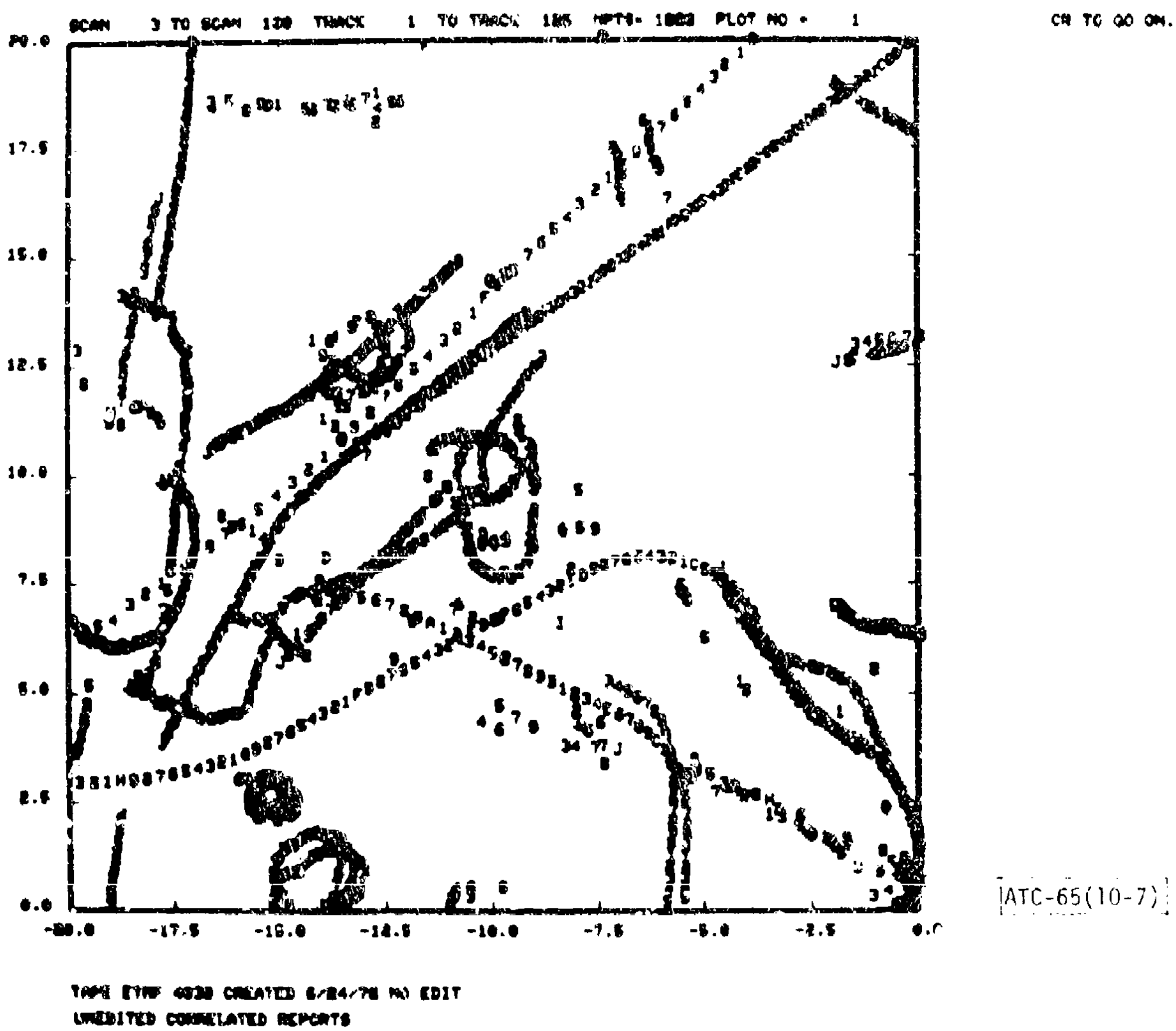
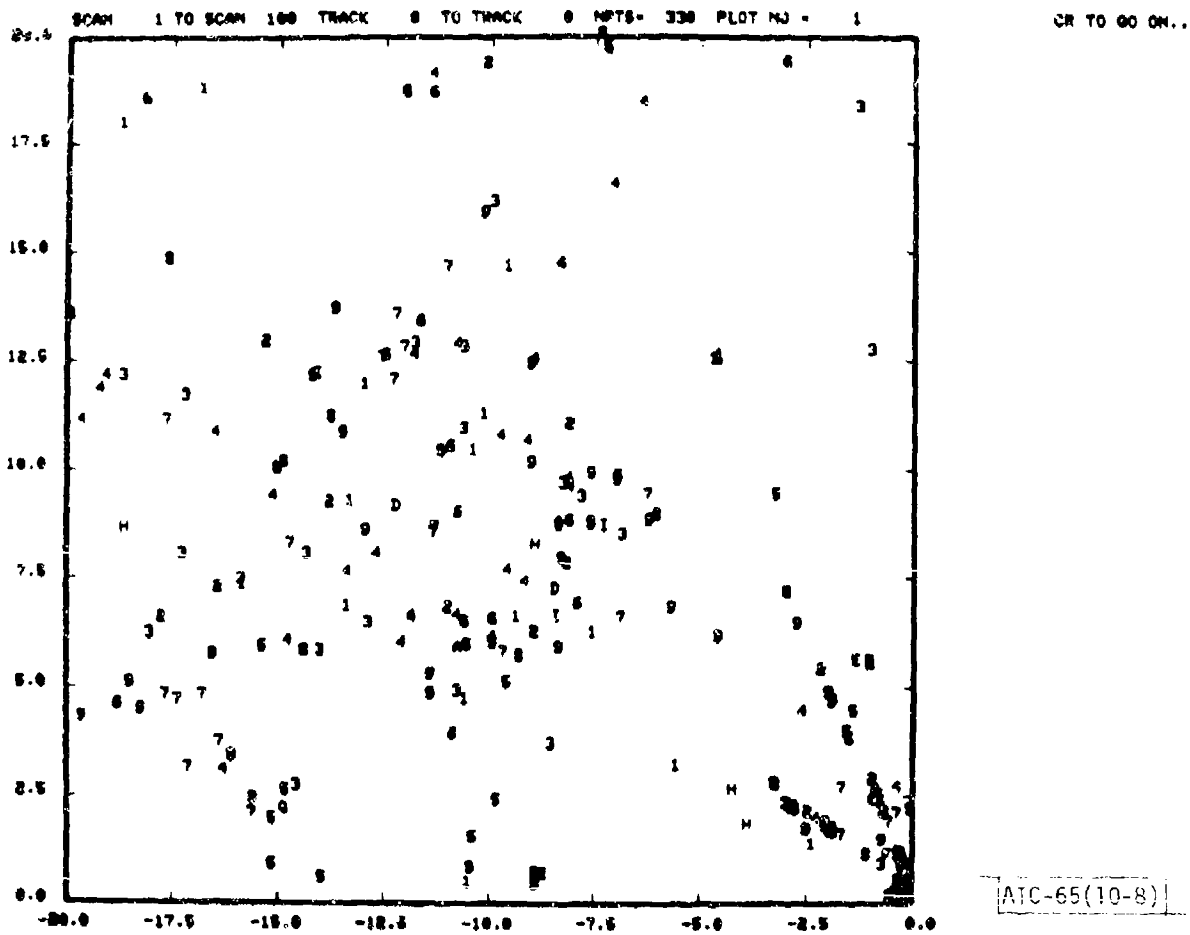


Figure 10-7: Unedited Correlated Reports

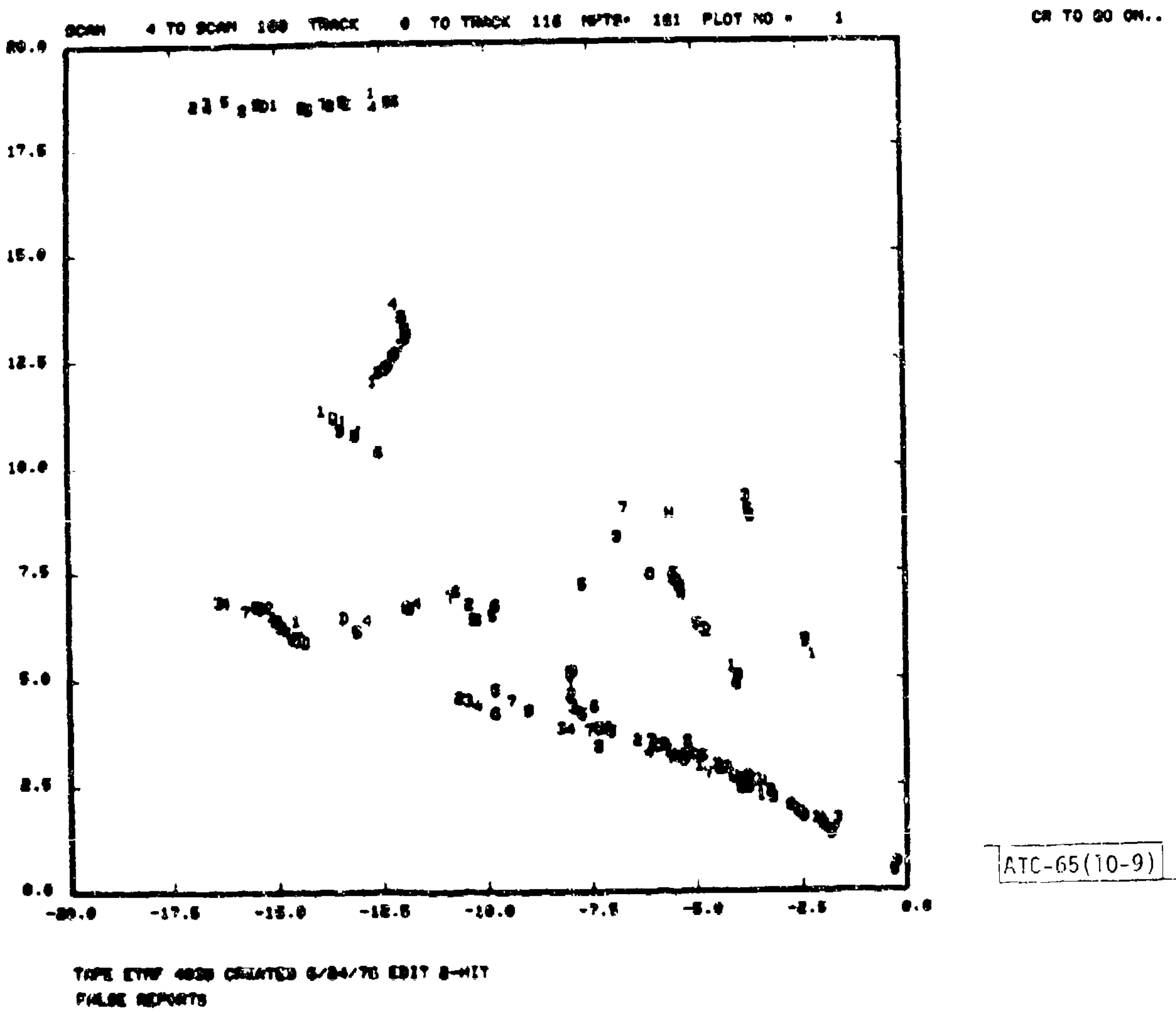
BEST AVAILABLE COPY



TAPE ETRF 4030 CREATED 6/24/76 EDIT 2-HIT
EDITED OUT REPORTS

ATC-65(10-8)

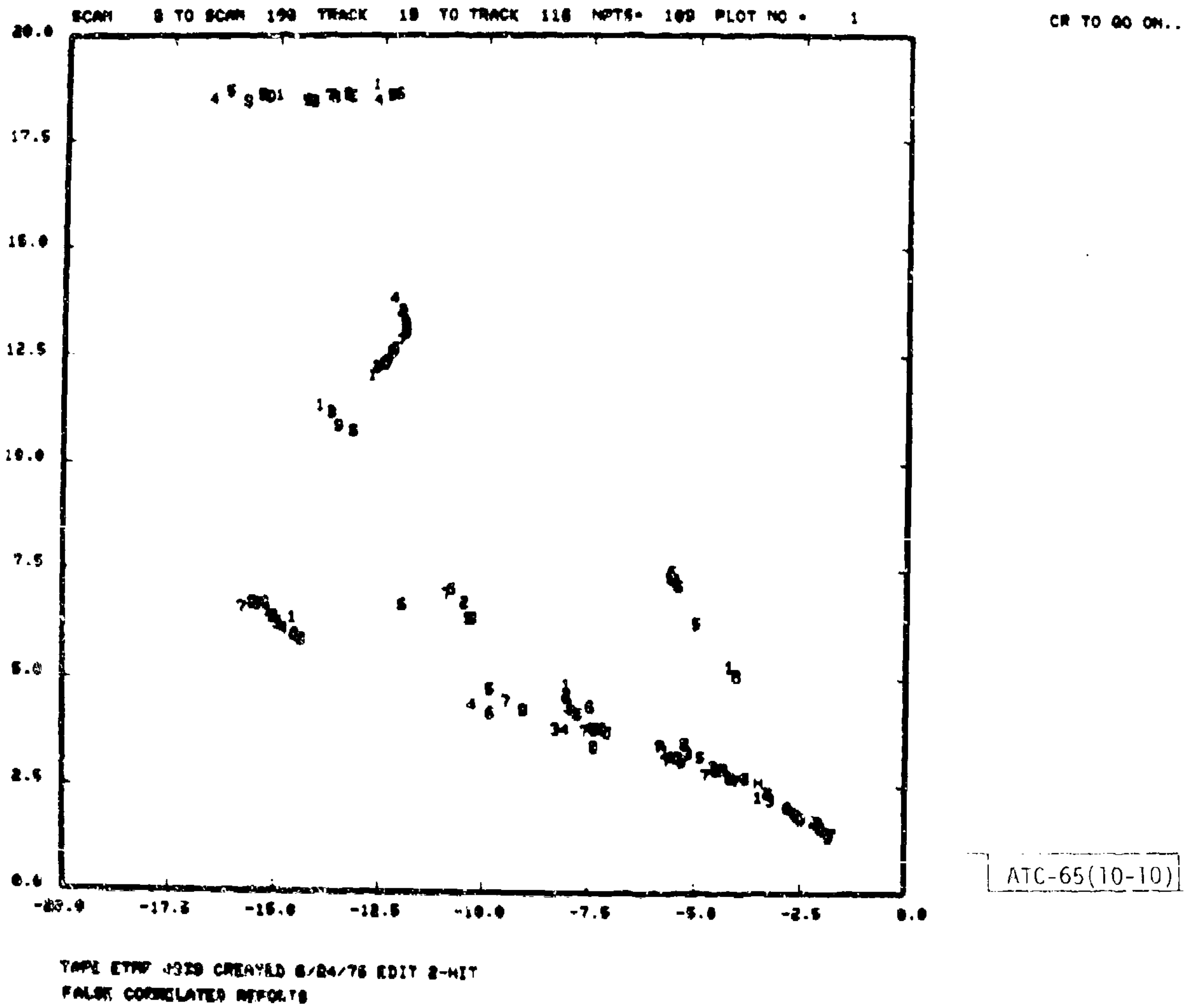
Figure 10-8: Edited Out Reports



ATC-65(10-9)

Figure 10-9: False Reports

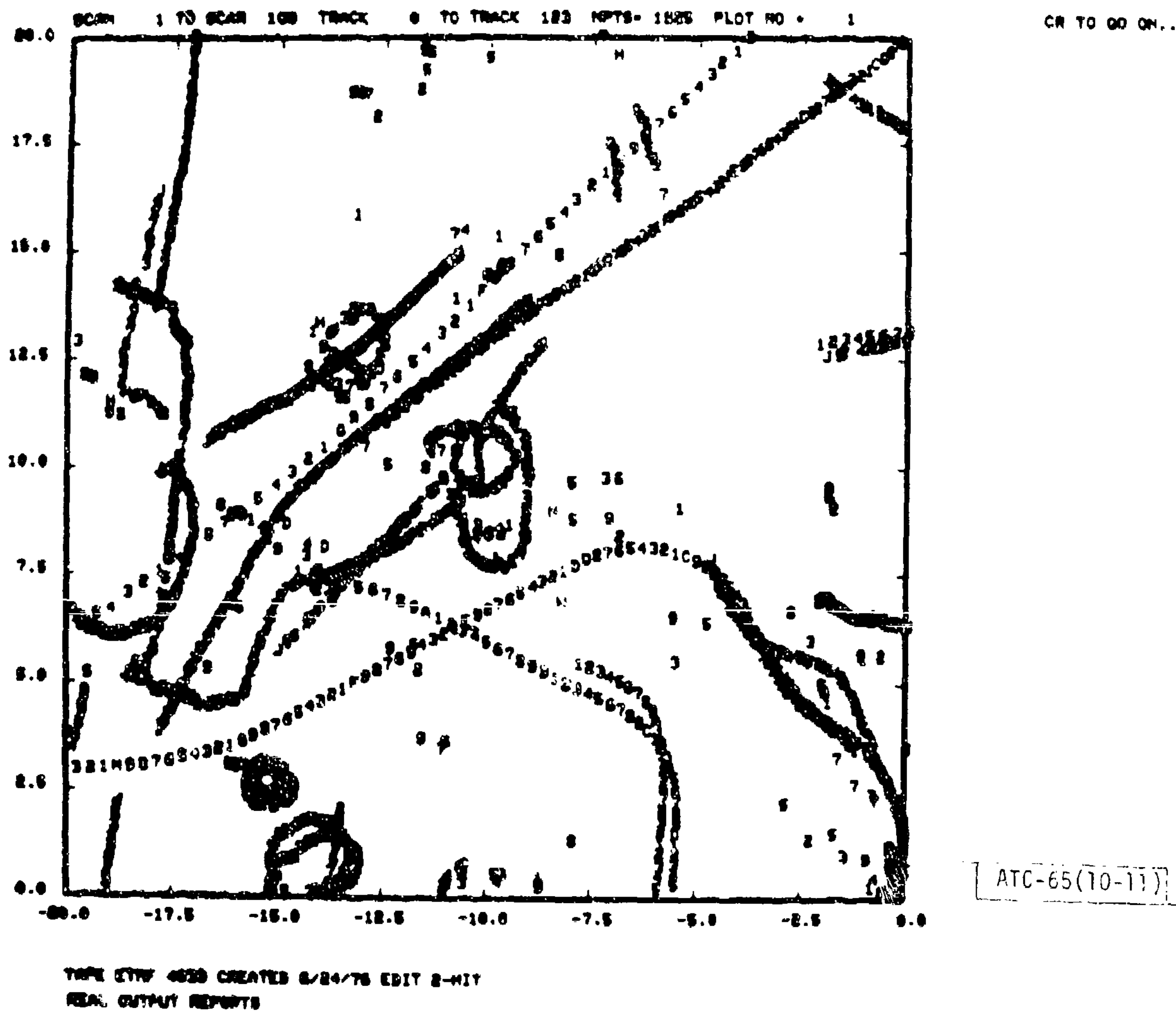
BEST AVAILABLE COPY



ATC-65(10-10)

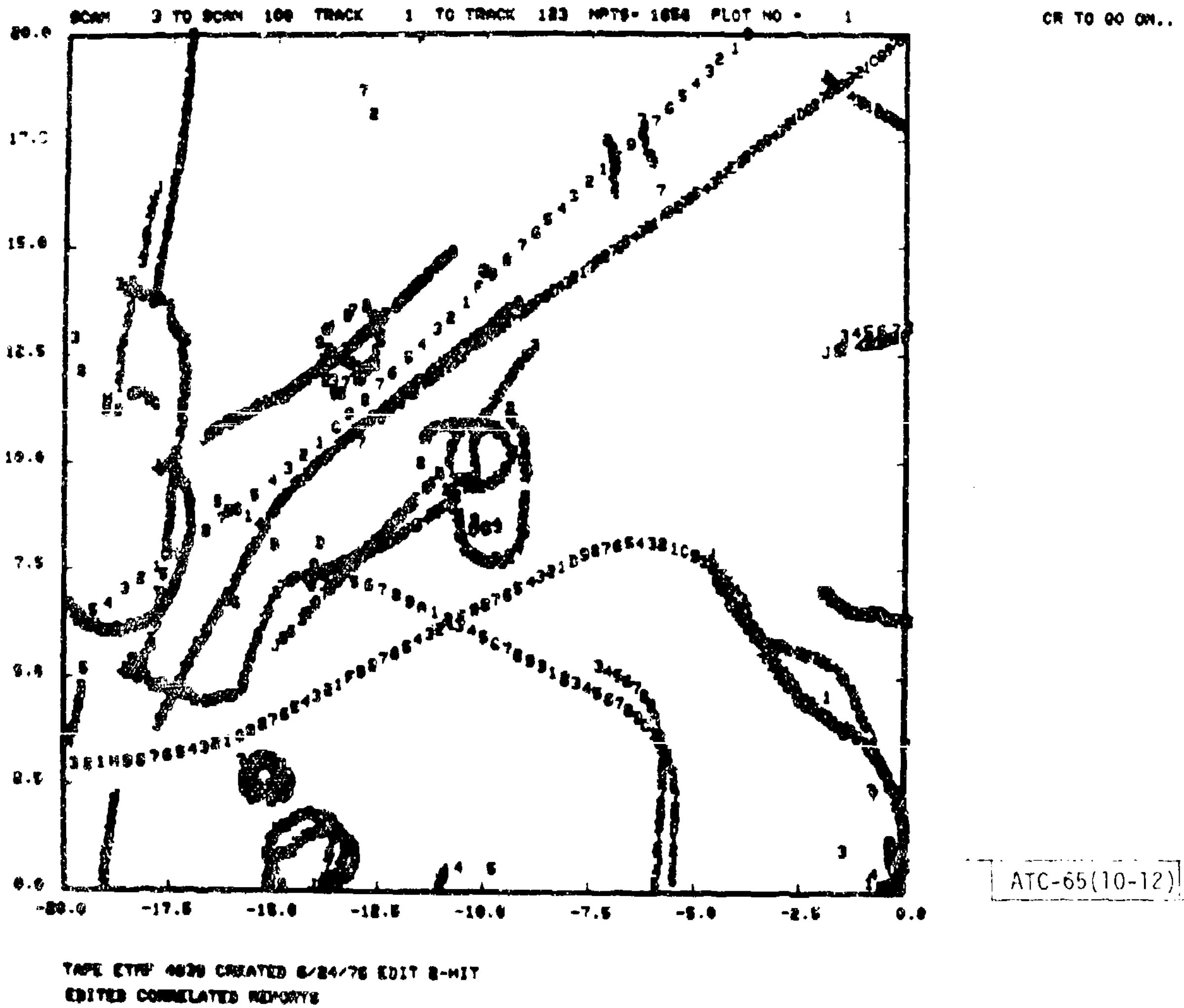
Figure 10-10: False Correlated Reports

BEST AVAILABLE COPY



ATC-65(10-11)

Figure 10-11: Real Output Reports



ATC-65(10-12)

Figure 10-12: Edited Correlated Reports

11.0 PRIMARY RADAR UTILIZATION

A fully equipped air traffic control sensor receives surveillance information from both beacon radar (ATCRBS) and primary radar interrogations. ATCRBS has the advantages of providing additional aircraft information (identity code and altitude) and being devoid of clutter, while primary radar provides coverage for shielded and nonbeacon-equipped aircraft and does not suffer degredation from reflection false targets. Thus, using both types of radar information jointly should provide optimum surveillance coverage.

An ATCRBS system that fully utilizes its primary radar information will use the radar reports for the following three functions:

1. Beacon reinforcement - beacon reports that correlate with radar reports are assumed to correspond to real aircraft rather than be due to fruit, reflection, or splitting

2. Beacon update - radar reports can be used to update beacon tracks when no beacon reports are received for them due to shielding or suppression.

3. Radar tracking - radar reports can be used to initiate and maintain tracks on aircraft that do not possess working beacon transponders.

It is clear that these functions require radar and beacon reports to be handled in unison. That is, separate radar and beacon algorithms cannot exist in the system, but rather, joint algorithms are required. Figure 11-1 presents a flowchart of the surveillance processing functional sequence that exists when radar reports are added to the ATCRBS system. It is assumed that both radar and beacon reports are received and processed one sector at a time, that both sets of reports have the same sector boundaries, and that both sets of reports are stored in report buffers prior to the start of the processing algorithms. These conditions imply that the radar and beacon antennas are collocated; a substantially more complex set of algorithms than those presented in this chapter are required if the antennas are physically separated.

The purpose of this chapter is to outline in detail how the existence and processing of radar reports fits into the algorithms described thus far in this paper. As will be seen, no major change is required in any of the routines that have been presented; only minor modifications are needed in order to incorporate the radar functions. In fact, very little software recoding would be required to add these functions to an ATCRBS system initially programmed to handle only beacon reports; each of the algorithms required by the radar processing was designed to be essentially the same as an algorithm used by the beacon system. If more than one feasible method was available to handle a radar function, the one chosen was the one that matched an existing beacon function. Thus, simple approaches were sometimes rejected in favor of more complex ones in order to simplify the overall joint system.

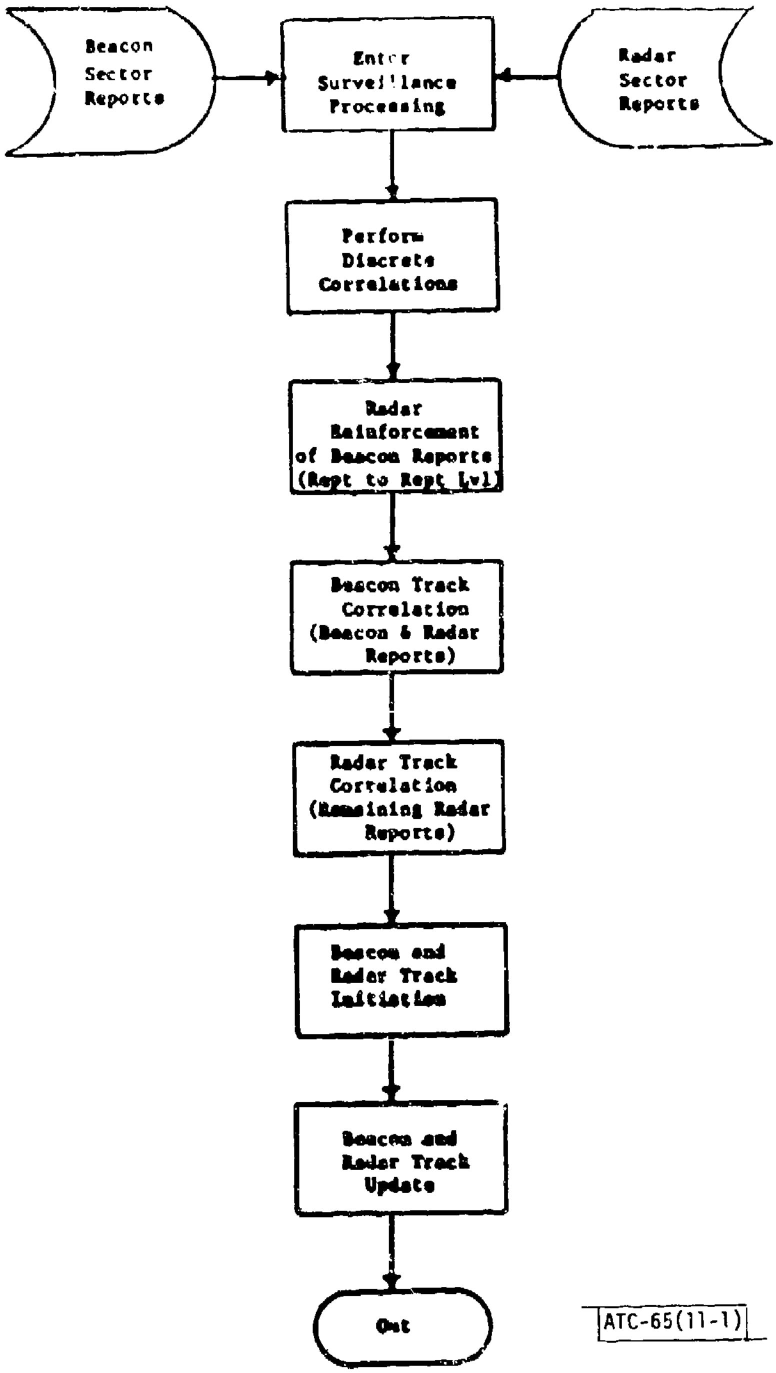


Figure 11-1: Radar Integration

It should be noted here that none of the radar algorithms to be presented have yet been tested. This is due to the fact that none of the new generation of moving target detection (MTD) radars are yet available for testing with DABS. Current radar systems (RVD) provide far too many false alarms to permit their use in the system discussed in this chapter. In particular, the number of radar only tracks that would be initiated by such radars would overwhelm the system capacity. It is quite possible that when real data from an MTD system becomes available, some changes in the algorithms described here will be required. It is being assumed, however, that such changes will be to parameters, equations, or scoring functions rather than to any fundamental concepts. A more detailed discussion of possible future modifications is contained in the last section of this chapter.

11.1 Radar Reinforcement

Most radar reports correspond to beacon-equipped aircraft. Thus the sensor will receive both a beacon and a radar report from these aircraft. The first radar processing function is to identify radar reports which are in essence duplicates of existing beacon reports. The beacon report in each such pair is marked as reinforced while the radar report is marked as "used" and is not allowed to participate in any subsequent processing function.

The basic idea of the reinforcement algorithm is the height of simplicity. A ρ, θ box is constructed around the position of each beacon report and all radar reports that fall within the box are identified. If no report is found, the beacon target is marked as unreinforced. If, on the other hand, one or more radar report is located, the nearest one is chosen as the reinforcer. The "distance" function applied in this calculation is defined as follows:

$$d = 100 \times \left[\frac{\Delta\rho}{\rho_{reinf}} + \frac{\Delta\theta}{\theta_{reinf}}\right]$$

where ρ_{reinf} and θ_{reinf} are the dimensions of the reinforcement box as depicted in Figure 11-2.

It should be evident that this reinforcement process is an exact analog of the target to track association and correlation processes described in Chapters 6 and 7. In particular, the following considerations arise:

1. A cross reference table of associating beacon and radar reports must be constructed.

2. Situations in which the reinforcement box straddles a sector boundary must be handled.

3. Intertwined situations in which two or more radar reports fall within the boxes of two or more beacon reports must be resolved.

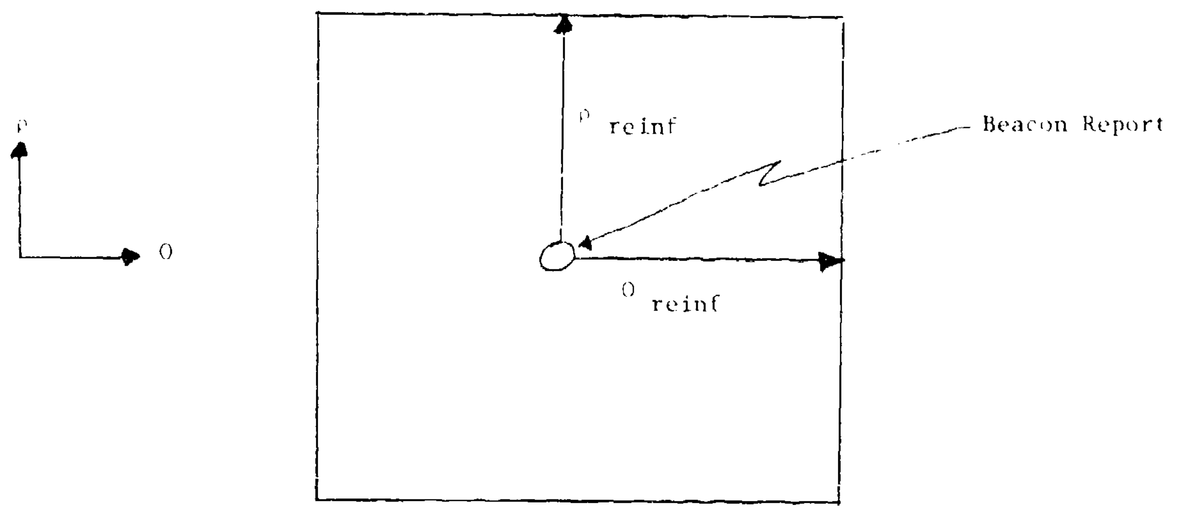


Figure 11-2: Radar Reinforcement Box

Thus, the most efficient way to handle the reinforcement algorithm is to use the program code and data structures previously developed for beacon correlation. Note that if this code didn't already exist, much simpler algorithms could be designed for radar reinforcement; however, it does exist.

Since all beacon reports have already been sorted by range (refer to Section 4.1), much execution time will be saved if the radar reports play the role of "tracks". Clearly, the same result is obtained if beacon reports are sought that fall within a box around a radar report instead of vice versa since the box size is independent of the report.

The reinforcement process commences by identifying all beacon reports that associate with each radar report. If radar report i has a range of ρ_i, all not yet reinforced beacon reports in the sector (see below) contained within sort bins

$$\frac{\rho_i - \rho_{reinf}}{\Delta\rho_{bin}} \text{ through } \frac{\rho_i + \rho_{reinf}}{\Delta\rho_{bin}} + 1 \text{ [integer division]}$$

are examined as being possible associants. The association is performed providing the two reports satisfy $\Delta\rho \leq \rho_{reinf}$ and $\Delta\theta \leq \theta_{reinf}$. For each pair so identified, an entry is made in the association cross reference table in the manner described in Section 6.2. A separate set of rows in the table, distinct from those used by beacon association, must be employed by this process to insure no beacon information will be overwritten. The score for the entry is equal to the distance measure defined above.

After all associations for the sector are determined, the reinforcement process follows the algorithm described for beacon target-to-track correlation. The only difference is that the Quality and Deviation Scores must be redefined to correspond to the different types of entities involved. For radar/beacon reinforcement, the Quality Score has very few attributes on which to base its association rating. In particular, radar reports have no code or altitude to match with those of the beacon report and only one association zone exists. Thus, the Quality Score is reduced to judging the certainty of the two reports corresponding to real aircraft. As shown in Figure 11-3, the beacon judgment is identical to that for the normal Quality Score, based on the hit pattern. The radar report attributes to use are presently undefined. The Deviation Score to be used for reinforcement is simply the "distance" score defined above. This value has already been calculated and is stored in the association cross reference table.

For each beacon/radar pairing that is determined, the beacon report is marked as reinforced and the radar report is marked as "used". If an unpaired beacon or radar report is found that is within $\Delta\theta_{reinf}$ of the sector boundary, the report is held over for processing in the subsequent sector. All other

Octal Digit & Factor	Condition	Score
7	unused	
6	unused	
5	unused	
4 Beacon hit pattern (modes A and C only)	3 or more replies	0
	2 replies of same mode	0
	1 reply of each mode	1
	1 reply	2
3 Radar Validity	not as yet defined	
2, 1, 0	Deviation Score	$100 \times \left[\frac{\Delta\rho}{\rho_{reinf}} + \frac{\Delta\theta}{\theta_{reinf}}\right]$

$$\text{Quality Score} = \left(d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0\right)_8$$

ATC-65(11-3)

Figure 11-3: Radar Reinforcement Quality Score

beacon reports are marked as unreinforced and all other radar reports enter into the functions described in the next two sections. Note that beacon reports that are held over until the next sector by the target to track correlation algorithm, but which are marked as unreinforced, do not enter into that sector's reinforcement process. Thus, a distinction exists between unreinforced and not yet reinforced beacon reports: the former have tried and failed, the latter are still trying.

ATCRBS and primary radar systems are subject to different false alarm mechanisms. Thus, a reinforced beacon report will almost always correspond to a real aircraft. This fact provides an additional mechanism for determining whether or not a suspicious beacon report is in fact a false alarm. There are three places in the ATCRBS algorithms presented in this paper where this knowledge is employed. First, digit 6 of the Quality Score (see Figure 7-1) is used to penalize suspect reports based on their hit pattern. If such a penalized report is reinforced, however, this penalty is removed. Furthermore, non-suspect reinforced reports are rewarded. The new definition of this digit thus becomes as shown in Figure 11-4.

The second change concerns the data editing function performed during track initiation. In that process, several classes of uncorrelated beacon reports are discarded as being false alarms. When radar information is available, this rule is modified so that it only pertains to non-reinforced reports. Finally, beacon reflection false targets will generally not be reinforced. Thus, more suspicion is cast when such a reinforced target is thought to be false. The image test is therefore modified such that a reinforced beacon target that passes the false criteria is labelled instead as possibly false, thereby reducing the likelihood of a real track ever being labelled as false.

Finally, the reinforcement algorithm itself requires one change in beacon target to track correlation. A track associating with a not yet reinforced target must be carried over to the subsequent sector before correlation is attempted (along with any other associating reports). This modification is needed for two reasons: to give the target a chance to be reinforced before being output, and to insure that the track correlates with the proper report (as reinforcement information is part of the Quality Score). Of course, if the track cannot be delayed for another sector for one of the reasons specified in Chapter 7, its correlation is permitted to proceed regardless.

11.2 Radar Update of Beacon Tracks

Occasionally no beacon report will be received for an aircraft even though it is beacon equipped. This could occur, for example, if the aircraft antenna were shielded from the sensor (such as during a turn), or if the aircraft transponder were temporarily suppressed, or if the aircraft flew

Octal Digit & Factor	Condition	Score
6	≥ 3, reinforced	0
number of replies (modes A and C only)	2 of same mode, reinforced	0
	1 of each mode, reinforced	1
	1, reinforced	2
	≥ 3, unreinforced	1
	2 of same mode, unreinforced	1
	1 of each mode, unreinforced	2
ATC-65(11-4)	1, unreinforced	3

Figure 11-4: Reinforcement Revised Quality Score

through a null of the beacon antenna. If primary radar reports are available to the sensor, it is likely that radar reports will exist for aircraft in these situations. In order to maintain accurate surveillance on these aircraft, it is desirable that the radar report be identified, correlated with the beacon track, and used to update its position.

Conceptually, the radar correlation procedure consists of attempting to match uncorrelated beacon tracks with unused (during reinforcement) radar reports. Since radar reports contain neither code nor altitude, positional nearness is the only available correlation criterion. As only uncorrelated beacon tracks are eligible to correlate with a radar report, it would appear that radar correlation must be attempted after beacon target to track correlation. The proposed method, however, identifies the correlating radar report for such tracks <u>during</u> the beacon correlation process.

This is accomplished by entering both beacon reports and unused radar reports into the association and correlation process at the same time. The scoring is arranged in such a way that radar reports cannot possibly be selected for correlation by a track unless no beacon report is available. In that event, however, the correlation process will select the proper radar report from among all contenders. Thus, both normal beacon correlation and radar correlation of beacon tracks are accomplished in one pass through the association and correlation algorithms presented in Chapters 6 and 7.

In order to perform this dual function, the unused radar reports must be added to the range sort table containing the beacon reports. The method for sorting each report is identical to that described for beacon reports in Section 5.1. With both beacon and radar reports sorted together, a track searching for associating reports will automatically find all reports of each type in one pass through the table.

The target-to-track association process checks for both identity code and altitude agreement between track and target. In order to force the association logic to perform in the desired manner, the following results are defined for radar report associations:

identity code check - disagreement

altitude check - potential agreement

This combined setting yields the following desirable effects:

1. All geometric zone 1 pairs are automatically associated

2. All geometric zone 2 pairs are further checked for velocity reasonableness

3. All geometric zone 3 pairs are discarded

4. No code swapping is attempted for radar reports

Thus, only "good" radar associations are permitted. All radar associations that are identified during this process are entered into the cross reference table in the identical manner used by beacon associations.

After association is completed, the correlation routine proceeds in exactly the same manner as described in Chapters 6 and 7. The only time that it even needs to know whether an association is radar or beacon is when it computes a Quality Score. The Quality Score values defined for radar associations are presented in Figure 11-5. As can be seen, the minimum Quality Score for a radar association is octal 47000000. Since the maximum score for a beacon association is 44773777, no radar association can be preferred over a beacon one. Thus, as stated earlier, only beacon tracks that fail to correlate with a beacon report can be updated by a radar report.

If a beacon track is to be correlated with a radar report, the correlation algorithm automatically selects the best one. Any intertwined or multiple association situations are resolved just as for beacon reports: Quality Scores consulted first, followed by Deviation Scores. Since the Deviation Score computation uses only position, it is directly applicable to radar reports as defined in Section 7.2.

The actual track update procedure for radar correlations is identical to that for beacon ones, except that, of course, no identity code or altitude update is possible.

11.3 Radar Tracking

Radar reports which correspond to neither beacon reports nor beacon tracks are generally due to the existence of non-beacon-equipped aircraft. Thus, in order to maintain surveillance on such aircraft, leftover radar reports must be entered into radar tracking algorithms. The set of such functions consists of radar track initiation, radar target to track correlation, and radar track update.

It is clear that the algorithms employed for the corresponding beacon functions can be used directly for radar processing. However, the absence of code and altitude in radar reports is expected to require more complex algorithms for adequate performance. Since the MTD radar data is not presently available, no detailed description of the "correct" radar algorithms can be provided at this time.

11.4 Possible Future Radar Modifications

The minimum information ever provided by a radar report is the range and azimuth of the illuminated aircraft. An MTD report, moreover, contains at least the following additional pieces of information:

Octal Digit & Factor	Condition	Score
7 dummy	none	4
6 dummy	none	7
5 Zone	Zone = 1 Zone = 2	0 1
4 radar validity	not as yet defined	
3 track validity	Same as Figure 7-1	
2, 1, 0 deviation score		

ATC-65(11-5)

$$\text{Quality Score} = (d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0)_8$$

Figure 11-5: Radar Association Quality Score

1. Amplitude

2. Doppler velocity

3. Number of returns making up the report

At present, a study is underway by the radar designers to determine what other pieces of information are available that would be useful for surveillance processing.

These designers are giving particular attention to the possibility of defining a radar report "code" and/or "altitude". Such attributes would be invaluable for radar only tracking. As employed by the target to track correlation algorithm, the definitions of these entities are as follows:

> "code" - a report attribute that should be nearly constant from scan to scan, but which is sufficiently changeable that failure to match cannot be used to rule out correlation
>
> "altitude" - a report attribute that cannot change by more than a fixed amount from scan to scan, so that a larger difference can be used to prevent correlation.

Thus, a radar report attribute that is characteristic of a particular aircraft, but which can suddenly change or be computed incorrectly in some circumstances would make a good "code"; an attribute that is variable within a known range would make an ideal "altitude".

If a "code" and "altitude" can be defined for radar reports, the radar tracking performance should equal that for beacon aircraft. The only program change required for radar in that case would be in the routines for determining "code" and "altitude" agreement, which would depend on the "code" and "altitude" specifications.

Unfortunately, if no good "code" or "altitude" exists for radar reports, the beacon algorithms would probably not perform adequately for radar targets. In particular, many false alarm tracks would be created and many track swaps could be expected. This is because position alone is insufficient in complex situations. In order to improve radar performance, several changes in the correlation and tracking algorithms are presently being studied. Some examples are:

1. Require three successive reports instead of two for track initiation

2. Don't report a radar track until it becomes established

3. Set a minimum track velocity to help eliminate cars, birds, etc.

4. If the resolution of a multiple association situation is not clear cut, "punt", and coast all involved tracks

5. Use a more sophisticated tracker than a two point interpolator

These alterations have been found to significantly improve radar tracking performance.

APPENDIX

A.0 ATCRBS MODE C ALTITUDE REPLY

When an aircraft equipped with an encoding altimeter is interrogated by a mode C transmission, it responds with a signal that contains an encoded version of its current altitude level. The code employed is a non-standard Gray code. As with all Gray codes, each encoded altitude level differs in only one bit position from the codes of the neighboring levels. This feature prevents erroneous readouts should the interrogation occur during an altitude change.

The ATCRBS mode C reply consists of twelve information bits, which can be grouped to form the four octal digits of the code employed. Thus, the ATCRBS altitude signal H can be written in either of the following two ways:

$$H = ABCD$$

$$= A_4A_2A_1B_4B_2B_1C_4C_2C_1D_4D_2D_1$$

where the subscripted letters correspond to bits and the non-subscripted ones to octal digits. The significance of the digits for the altitude reply is altered from this normal order. Specifically, the ordering employed is

DABC

That is, the C digit varies most rapidly, the D least rapidly.

In all Gray codes, the sequence of values generated by each digit is reflexive about the end values. That is, they count up, then down, then up, and so forth. For example, the sequence of values assumed by the A or B octal digit is given by:

0, 4, 6, 2, 3, 7, 5, 1, 1, 5, 7, 3, 2, 6, 4, 0, 0, 4

The C digit employs a subset of this sequence, namely:

4, 6, 2, 3, 1, 1, 3, 2, 6, 4, 4, 6, . . .

Finally, the D digit only uses the truncated set of values

0, 4, 6, 2

as this range is sufficient to cover all altitude levels of interest.

At the time a digit repeats its end value (0 or 1 for A or B, 4 or 1 for C), the next more significant digit proceeds to its next value. Thus, one full cycle of values for any digit corresponds to two values of the next higher digit: one value during the ascent stage and the other during the return. From this set of information, it is possible to calculate the period of each octal digit, that is, the number of flight levels required for one complete sequence through all values. The results are:

period of C: 10 values x 1 level per change = 10

period of B: 16 values x 1/2(10) levels per change = 80

period of A: 16 values x 1/2(80) levels per change = 640

The 1/2 factor in the latter two calculations is required because a digit changes twice during the cycle of the next lower digit.

Using the facts developed above, it is possible to decode a mode C reply and determine the flight level it represents. Also, by reversing the process, a given flight level can be encoded into its bit pattern. The former procedure is used to enter the aircraft altitude into a target report, while the latter one is sometimes required in target to track association. The next two sections of this appendix will present the algorithms employed to convert from one form to the other. The remainder of the appendix will describe how altitude information is employed in various places in the surveillance processing system.

A.1 Encoding Algorithm

Since the encoding algorithm, which converts flight level into Gray code, is easier to understand and serves to motivate the decoding process, it will be presented first. The simplest encoding procedure, of course, would be to perform a table lookup for the given flight level. However, since over 1000 entries would be required for the table, this approach was rejected as not being cost effective.

The algorithm selected follows directly from the period calculations of the previous section. It determines, through use of modulo arithmetic, how far into each octal digit's sequence the given altitude level falls. Then, knowing the actual value sequence employed by each digit, the correct encoded value for the digit can be identified. Finally, the four individual digit values are weighted properly to construct the code word.

For each digit i(C=1, B=2, A=3, D=4), define the following two quantities:

P_i = period of digit i (calculated in A.0)

L_i = number of levels per change of digit i (L_1=1, L_i=P_{i-1}/2)

Each digit i repeats its value sequence every P_i flight levels. Thus, if the flight level to be encoded is reduced modulo P_i, the relative position within a period is determined. That is:

$$\hat{H}_i = H \mod P_i$$

where

$$H_i = \text{flight level}$$

$$H' = H + 12$$

$$\hat{H}_i = \text{relative position within period of digit i.}$$

The addition of 12 is required because the lowest encoded flight level is -12, not 0. Once $\hat{H}_i$ is known, the required element of the digit i sequence is found simply as:

$$d_i = \frac{\hat{H}_i}{L_i} \quad \text{integer division}$$

Knowing the sequence employed, the proper value can be selected.

The details of the overall encoding algorithm are presented in Figure A-1, while Figure A-2 presents the calculations for a sample altitude level.

A.2 Decoding Algorithm

The decoding algorithm, which converts the Gray coded altitude representation into the integer flight level, is in essence the inverse of the previous procedure. Again, a straight table lookup would be the easiest algorithm, but a 4096 element table would be required. Thus, the small increase in processing time needed by the process to be described here was felt to be a good storage/time tradeoff.

The algorithm employed first breaks down the code word into its octal digits. Then, knowing the sequence of values assumed by each digit, and the number of levels between each change of the digit, is it possible to calculate the contribution of each digit. The desired flight level is finally the sum of all the individual digit contributions.

The only complication that arises in this procedure is that the sequence of values for any digit is double valued, each value appearing in both the ascent and return stages. The correct choice to utilize can only be determined if all of the more significant digits have been processed first. Thus the digit contributions, unlike for normal counting systems, are not independent of each other.

INPUT: decimal flight level H

ATC-65(A-1)

CALCULATIONS:

$$H' = H + 12$$

$$C = T_1 \left(\left[H' - \left(\frac{H'}{10}\right) \times 10 \right] \right)$$

$$B = T_2 \left(\left[H' - \left(\frac{H'}{80}\right) \times 80 \right] / 5 \right)$$

$$A = T_2 \left(\left[H' - \left(\frac{H'}{640}\right) \times 640 \right] / 40 \right)$$

$$D = T_2 \left(H' / 320 \right)$$

all divisions are integer division

(no remainder)

T_1 (0-9) : 4, 6, 2, 3, 1, 1, 3, 2, 6, 4

T_2 (0-15) : 0, 4, 6, 2, 3, 7, 5, 1, 1, 5, 7, 3, 2, 6, 4, 0

OUTPUT: $W = A \times 2^9 + B \times 2^6 + C \times 2^3 + D$

Figure A-1: Flight Level to Gray Code Algorithm

INPUT: $H = 73$

CALCULATIONS:

$$H' = 73 + 12 = 85$$

$$C = T_1\left(\left[85 - \left(\frac{85}{10}\right) \times 10\right]\right) = T_1(85 - 8 \times 10) = T_1(5) = 1$$

$$B = T_2\left(\left[85 - \left(\frac{85}{80}\right) \times 80\right] / 5\right) = T_2([85 - 1 \times 80] / 5)$$

$$= T_2(5/5) = T_2(1) = 4$$

$$A = T_2\left(\left[85 - \left(\frac{85}{640}\right) \times 640\right] / 40\right) = T_2([85 - 0 \times 640] / 40)$$

$$= T_2(85/40) = T_2(2) = 6$$

$$D = T_2(85/320) = T_2(0) = 0$$

OUTPUT:

$$W = 6 \times 2^9 + 4 \times 2^6 + 1 \times 2^3 + 0$$

$$= 6410_8$$

$$= 110100001000$$

$$= A_4 A_2 A_1 B_4 B_2 B_1 C_4 C_2 C_1 D_4 D_2 D_1$$

Figure A-2: Example Use of Encoding Algorithm

The straightforward approach to the decoding process would thus proceed as follows. First determine the contribution of the most significant digit, D. Once D is known, the phase of the A digit can be determined and then its contribution computed. Similarly process the B digit, and finally the C digit. This procedure would require four table lookups and three phase calculations.

The actual implementation that has been chosen reduces this complexity to three table lookups and one phase calculation at the cost of a slight increase in storage. The suggested algorithm is presented in Figure A-3, the tables required are given in Figure A-4, and a sample application is illustrated by Figure A-5.

The algorithm begins by identifying the joint AB and individual C and D values by the indicated shifting and masking operations. Next, the combined AB value is used as an index into the T_{AB} table. This table provides the position that this value occupies in the joint AB sequence under the assumption that A is in its ascending phase (this assumption is checked after D is processed). In addition, if the entry has the hundreds digit set, it marks the C digit as ascending; if not, as returning. For example, if $AB = 33_8 = 27_{10}$, $T_{AB}(27) = 136$ indicates that this value is the 37th in the joint value sequence (0,1,...,36) and that the C digit should be processed as ascending.

The AB contribution is then found by multiplying the sequence position by 5 levels per positions, after which the C contribution is included. The T_C table gives the contribution of C if it is ascending. Thus, by the reflexive nature of the Gray code, $4-T_C(C)$ is the contribution for a returning C. Finally, the contribution of D is found, and the phase of A is checked. If A is ascending, the calculation is finished; if not, the ABC contribution is corrected by using the reflexive nature of the code once again.

A.3 Target and Track Altitude Representations

Depending upon the sophistication of its equipment, an aircraft can respond in one of three ways to a mode C interrogation:

1. Send a reply containing an encoded altitude signal of the form discussed above,

2. Send a reply containing only bracket pulses,

3. Send no reply at all.

The second category indicates the absence of an operational encoding altimeter, while the third one indicates a minimal transponder.

INPUT: encoded altitude G (12 bits)

CALCULATIONS:

$$AB = \frac{G}{8^2}$$

$$C = \frac{G - AB \times 8^2}{8}$$ integer divisions

$$D = G - AB \times 8^2 - C \times 8$$

$$TEMP = T_{AB}(AB) \times 5$$

IF (TEMP > 500) THEN NEXT = TEMP + T_C (C) - 500

ELSE NEXT = TEMP + (4 - T_C(C))

$$THIRD = T_D(D) \times 320$$

IF (THIRD = 0 or 640) THEN FL = THIRD + NEXT

ELSE FL = THIRD + (319 - NEXT)

OUTPUT: H = FL - 12

(see Figure 4 for T_{AB}, T_C, and T_D)

Figure A-3: Gray Code to Flight Level Algorithm

TABLE $T_{AB}(i)$, i = 0, 63

$T_{AB}(0)$ =	100	7	3	104	1	106	102	5
$T_{AB}(8)$ =	63	156	160	59	162	57	61	158
$T_{AB}(16)$ =	31	124	128	27	130	25	29	126
$T_{AB}(24)$ =	132	39	35	136	33	138	134	37
$T_{AB}(32)$ =	15	108	112	11	114	9	13	110
$T_{AB}(40)$ =	148	55	51	152	49	154	150	53
$T_{AB}(48)$ =	116	23	19	120	17	122	118	21
$T_{AB}(56)$ =	47	140	144	43	146	41	45	142

TABLE $T_C(i)$, i = 0, 7

$T_C(0)$ =	0	4	2	3	0	0	1	0

TABLE $T_D(i)$, i = 0, 7

$T_D(0)$ =	0	0	3	0	1	0	2	0

Figure A-4: Tables for Decoding Algorithm

INPUT: $G = 7064_8 = 111000110100$

CALCULATIONS:

$$AB = \frac{7064_8}{100_8} = 70_8 = 56$$

$$C = \frac{7064_8 - 7000_8}{10_8} = 6_8 = 6$$

$$D = 7064_8 - 7000_8 - 60_8 = 4_8 = 4$$

$$TEMP = T_{AB}(56) \times 5 = 47 \times 5 = 235$$

$$TEMP < 500$$

$$\therefore \quad NEXT = 235 + (4 - T_C(6)) = 235 + 4 - 1 = 238$$

$$THIRD = T_D(4) \times 320 = 1 \times 320 = 320$$

$$THIRD \neq 0 \text{ or } 640$$

$$\therefore \quad FL = 320 + (319 - 238) = 401$$

OUTPUT: $H = 401 - 12 = 389$

Figure A-5: Example of Decoding Algorithms

If a reply is sent, it can be received by the sensor as clear or garbled, depending upon the aircraft and fruit environment. It is categorized as clear if the reply processor declares all its bits as high confidence, and garbled if any low confidence bits exist. Thus, the altitude that is entered into a target report can be any of the following five classifications:

1. Unknown (no replies received)

2. Garbled brackets only

3. Clear brackets only

4. Garbled flight level

5. Clear flight level

If case 5 exists for a report, the code bits are decoded by the algorithm presented in Section A.2 and the integer flight level is placed into the report altitude field.

The manner in which each of these five types of altitude information is represented in a target report is depicted in Figure A-6. Remember that in this implementation both the code and confidence words of any mode consist of 16 bits: the 12 information bits, followed by F1, F2, X, and SPI. Since neither X or SPI is used on mode C, and since the F1 and F2 values are immaterial, the four "appendage" code and confidence bit positions are free to be used for other purposes. As shown in Figure 3-2, these confidence positions contain the altitude type setting defined in Figure A-6.

The altitude contained in a track file, since it is built from those of the constituent reports, could be any of these same five types. In addition, though, several more track altitude classifications are required because of the following rule expressed in Section 9.2:

> If the altitude of a track has not been updated for 3 (parameter) scans, set all altitude bits to low confidence.

This rule is intended to prevent the rejection of an association due to out-of-date altitude information.

The result of this confidence word modification is that altitude classifications of the type "had been X", where X is one of the five forms presented above, are required. The expanded list of track altitude types and their track file settings (see Figure 8-6) is given by Figure A-7. The two possible categories "had been clear brackets" and "had been garbled brackets" have been collapsed into the single category "had been brackets", as all code bits in either case are believed to be zeroes.

Code or confidence word bit ordering:

$\underbrace{A_4A_2A_1B_4B_2B_1C_4C_2C_1D_4D_2D_1}_{\text{12 code bits}}\underbrace{F_1F_2X\ SPI}_{\text{4 Appendage bits}}$

ATC-65(A-6)

	Altitude Type	Code	Confidence	Type Setting
1.	no replies	000	FFF	2
2.	garbled brackets	no high confidence '1' s, at least 1 high confidence '0', and at least 1 low confidence bit		3
3.	clear brackets	000	000	4
4.	garbled flight level	all bits low confidence; or at least 1 high confidence '1' and at least 1 low confidence bit		1
5.	clear flight level	any	000	0

All values in Hex

Figure A-6: Report Altitude Representations

	Altitude Type	Code	Confidence	Type Setting
1.	no replies	000	FFF	2
2.	garbled brackets	No H1, $\geq$1 H0, $\geq$1 low conf.		3
3.	clear brackets	000	000	4
4.	garbled flight level	all low conf.; or $\geq$1 H1 and $\geq$1 low conf.		1
5.	clear flight level	any	000	0
6.	"had been" no replies	000	FFF	A
7.	"had been" brackets	000	FFF	D
8.	"had been" garbled flight level	any	FFF	E
9.	"had been" clear flight level	any	FFF	F

ATC-65(A-7)

All values in Hex

Figure A-7: Track Altitude Representations

A.4 Target-to-Track Altitude Association

One of the criteria used to rate a potential association between a target report and a track, as discussed in Section 6.2, is the degree of compatibility that exists between the altitudes of the two entities. The variable Δh_{ij} is used to represent the difference in flight levels between the altitudes of track i and report j. The interpretation given to various values of Δh_{ij} is as follows:

$$0 \leq \Delta h_{ij} \leq \frac{1}{2} \Delta h_{max}\text{: agreement}$$

$$\frac{1}{2} \Delta h_{max} < \Delta h_{ij} \leq \Delta h_{max}\text{: potential agreement}$$

$$\Delta h_{ij} > \Delta h_{max}\text{: disagreement}$$

Typically, Δh_{max} is set at 10 flight levels, or 1000 feet.

The first requirement of altitude agreement is that the track and target represent the same type of aircraft. That is, both must be no replies, or both brackets only, or both flight level. If either of the first two of these are found to be the case, the result automatically becomes $\Delta h_{ij} = 0$. If both target and track represent an altitude reporting aircraft, however, further checking is required.

In the simplest case, both target and track will have the altitude classification clear flight level. Since both altitudes will then be stated in integer flight levels, a subtraction will directly yield the difference between them. The per scan difference, which is the critical value, is thus given by:

$$\Delta h_{scan} = \frac{|\Delta h|}{S} \qquad \text{integer division}$$

where S is the number of scans since the track altitude was updated. If this difference is no greater than Δh_{max}, Δh_{ij} is set equal to the difference. The magnitude of Δh_{ij} will than indicate whether agreement or potential agreement applies.

However, if the difference exceeds Δh_{max}, or if one or the other of the altitudes is garbled flight level, a more complex procedure is required. First, the clear altitude (or both in the case of the subtraction failure) is converted back into its encoded representation by the algorithm presented in Section A.1. Then the high confidence bits of the two encoded representations are compared with each other. Should the track altitude be of type "had been flight level (clear or garbled)", all confidence bits are assumed to be high for this test; problems caused by this action are corrected below.

According to the discussion of Section A.0, the altitude code digits DAB, taken as a group, change their value every 5 flight levels. Thus, if the two encoded altitudes have no high confidence bit differences among the DAB bits, they could be from 0 to 4 levels apart. Similarly, if they differed from each other only in the <u>correct</u> bit, they could be from 5 to 9 levels apart, and so forth. The algorithm that has been implemented does not determine whether or not the correct bit is the one affected when the two altitudes differ in one bit among DAB, as the determination would be too complex to justify. Instead, it assumes such is the case. Thus, the value given to Δh_{ij} as a result of the bit comparison is calculated as:

$$\Delta h_{ij} = \left[5 + 5 * \text{Max}\left\{0,\ d_{high} - N_e\right\}\right] / S$$

where

d_{high} = number of high confidence bit differences in DAB

N_e = number of bit errors assumed possible in the reply processor

The fixed value of 5 is intended to account for the uncertainty provided by the low confidence bits of the altitudes. To this figure, additional increments of 5 are added for each bit difference that cannot be accounted for by reply processor errors. Clearly, depending upon the number of such differences, the result of the comparison could be altitude agreement, potential agreement, or disagreement.

In the event the target and track represent different types of aircraft, fixed values of Δh_{ij} are assigned to the potential association. In each case, the result will be placed into the potential agreement category. This is done to reflect the possibility of an aircraft changing its type of response. For example, in a fade it is possible that no mode C replies will be received at the sensor, or the mode C replies could be blocked by synchronous garble or other effects. Also, it is conceivable that an aircraft will turn its encoding altimeter on or off during flight, thus converting from flight level to brackets only, or vice versa. The actual values assigned to mixed associations are determined by the fractional parameters P_{h1} and P_{h2} as follows:

$\Delta h_{ij} = P_{h1} * \Delta h_{max}$ if either target or track has no replies

$\Delta h_{ij} = P_{h2} * \Delta h_{max}$ if the association is brackets only versus flight level

Nominally, $P_{h1} = .9$ and $P_{h2} = .8$.

The one exception to this rule occurs if the mixed association is garbled brackets versus flight level (clear or garbled). Since garbled brackets could actually be garbled flight level, an attempt is first made to compare on that basis. If the result of the bit pattern comparison scores better than $P_{h2}*\Delta h_{max}$, that result is accepted instead.

After Δh_{ij} has been computed by the applicable rule presented above, one final step remains. The track classifications "had been X", as explained in the previous section, are used to indicate that the track altitude information is out of date. Thus, no association will be allowed to be rejected with such a track due to altitude mismatch. If Δh_{ij} exceeds Δh_{max} for a track in one of the "had been X" categories, the value is automatically lowered to Δh_{max}.

A complete summary of the various procedures used to compute Δh_{ij} for all possible target report versus track cases is presented in Figure A-8. The five report and nine track classifications shown in the table were all defined in Section A.3.

A.5 Track Altitude Update

Once per scan, each track file in the system is updated in the manner described in Chapter 9. This section will describe the rules employed in the update of the altitude and altitude confidence fields.

If a track correlates with a target report on the current scan, and if the altitude of the report is acceptable (as defined below), the track altitude fields are updated by the report altitude information. However, if neither condition is satisfied, the track altitude in essence "coasts". To prevent the information from becoming too old to be of any value, a two-phase timeout procedure is utilized.

Corresponding to each system track is an altitude counter. This counter is zeroed every time the altitude fields are successfully updated by a new report. If no update is possible, the counter is incremented. When its value reaches a parametric number of scans (nominally 3), the altitude confidence field of the track is set to indicate all altitude bits low confidence. Thus, the track becomes a member of one of the "had been X" classifications described in Section A.3. This setting maintains the most recent altitude information known for the track so that potential associations may be scored properly. However, as described in the previous section, no association may be rejected for a track in this state.

Should altitude update failures continue after this point, the altitude counter is decremented one unit per scan. When it reaches zero, the track altitude information is defined to be useless. Thus, the next time the track correlates, the altitude and altitude confidence fields of the report are automatically placed into the track file. Then the entire sequence begins again.

Track \ Report	(1) no replies	(2) garbled brackets	(3) clear brackets	(4) garbled level	(5) clear level
(1) no replies	0	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$
(2) garbled brackets	$P_{h1}*\Delta h_{max}$	0	0	Compare bits or $P_{h2}*\Delta h_{max}$	Compare bits or $P_{h2}*\Delta h_{max}$
(3) clear brackets	$P_{h1}*\Delta h_{max}$	0	0	$P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$
(4) garbled level	$P_{h1}*\Delta h_{max}$	Compare bits or $P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$	Compare bits	Compare bits
(5) clear level	$P_{h1}*\Delta h_{max}$	Compare bits or $P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$	Compare bits	$\|diff\|$ or Compare bits
(6) "had been" no replies	0	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$	$P_{h1}*\Delta h_{max}$
(7) "had been" brackets	$P_{h1}*\Delta h_{max}$	0	0	$P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$
(8) "had been" garbled level	$P_{h1}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$	Compare bits or Δh_{max}	Compare bits or Δh_{max}
(9) "had been" clear level	$P_{h1}*\Delta h_{max}$	Compare bits or $P_{h2}*\Delta h_{max}$	$P_{h2}*\Delta h_{max}$	Compare bits or Δh_{max}	$\|diff\|$ or Compare bits or Δh_{max}

Δh_{ij} computation or value is given

or means choose best score

ATC-65(A-8)

Figure A-8: Altitude Association Cases

The rule that governs the acceptability of a correlating report's altitude can be expressed as follows:

> A report altitude can be used to update a track file only if it agrees with the current track altitude (i.e: $\Delta h_{ij} \leq \frac{1}{2} \Delta h_{max}$) and it has at least as good quality as the current altitude.

The first clause of the rule is straight-forward. It is meant to prevent incorrect correlations from invalidating the authenticity of the track file information. The second clause means that garbled altitude information may not replace a clear flight level. If this were done, the position of the aircraft would become unknown, as garbled altitude cannot be decoded.

The overall 9x5 update acceptability matrix is presented in Figure A-9. Again, the classifications are those defined in Section A.3. Entries labelled unacceptable mean that the altitude counter progresses in the manner described above. Those labelled replacement mean that the target report altitude fields replace those currently in the track file, and the altitude counter is zeroed. Finally, if both the target and track are garbled brackets, the track altitude confidence field is improved by setting to high confidence all currently low confidence bits that are high confidence in the report. Note that this improvement rule is not employed if both the track and report are garbled flight level. To do so could result in a flight level being produced that is wildly different from that at which the aircraft actually resides.

ATC-65(A-9)

Report \ Track	(1)	(2)	(3)	(4)	(5)
	Same labels as A-8				
(1)	R	U	U	U	U
(2)	U	I	R	U	U
(3)	U	L	R	U	U
(4)	U	U	U	U	R
(5)	U	U	U	U	R
(6)	R	D	D	D	D
(7)	D	D	R	D	D
(8)	D	D	D	D	R
(9)	D	D	D	D	R

Same labels as A-8 (rows)

altitude update actions:

R - if $\Delta h_{ij} \leq \frac{1}{2} \Delta h_{max}$, replace track altitude with report altitude and zero counter
if $\Delta h_{ij} > \frac{1}{2} \Delta h_{max}$, proceed as under U (or D)

U - increment counter; if reach parametric value, set track to "had been" category

D - decrement counter; if reach 0, replace track altitude with report altitude

I - improve track altitude by union of high confidence bits

L - set counter to zero, leave track altitude as is

Figure A-9: Altitude Update Cases